Proceeding of the IEEE
International Conference on Robotics and Biomimetics
Dali, China, December 2019

Rapidly-exploring Random Trees Based on Cylindrical Sampling Space

Fujie Yu, Yuan Chen, Xiao Zhang, Xilong Xin*
School of Mechanical, Electrical & Information Engineering, Shandong University at Weihai, China
Mail: cyzghysy@sdu.edu.cn

***Abstract* - In this paper, the defect of traditional fast-expanding random tree (RRT) is optimized for the path planning in three-dimensional environment. Due to the randomness generated by its own nodes, the traditional RRT algorithm has the defects of the planned path twist and easy to cross the target point, and greatly differs from the ideal path, so that the time required for the robot to implement path tracking is greatly increased. In this paper, an RRT algorithm based on cylindrical sampling space is proposed. The algorithm uses the line connecting the starting point and the ending point of the path as the central axis. The maximum distance from all nodes on the path to the cylinder axis is taken as the radius value of the first cylinder. The cylinder space is used as the new node sampling space in the next path planning. Then, a new cylinder is generated by reducing the radius of the previous cylinder as the current sample sampling space. Repeat the above steps until find the last optimized path. The path generated by the traditional RRT algorithm and the path generated by the RRT optimization algorithm based on cylindrical sampling space are compared and analyzed under the given obstacle and obstacle-free environment. The simulation results show that the path generated by the RRT optimization algorithm based on the cylindrical sampling space is smoother and shorter, almost no redundant path. The path meets the actual motion requirements of the robot.**

Index Terms - Rapidly expanding random trees; path plan; Node generation space.

I. Introduction

For the path planning in the three-dimensional environment, some scholars have proposed different solutions. Traditional path planning algorithms include grid method[1], artificial potential field method[2], polygon fitting method, genetic algorithm[3], etc., but these methods need to model obstacles in the determination space, which is not suitable for solving path planning in complex environments[4-5]. Based on the fast-expanded random tree (RRT) path planning algorithm, collision detection of sampling points in space avoids the modeling of obstacles, and can effectively solve the path planning problem with complex constraints[6]. However, although RRT algorithm can avoid obstacles in three-dimensional environment and realize fast path planning, due to the randomness generated by its own nodes, the planned paths are often very tortuous and easy to cross the target point, which is very different from the ideal path. The time required for the robot to achieve path tracking is greatly increased.

Aiming at the defects of traditional RRT algorithm in robot path planning, this paper proposes an RRT optimization algorithm based on cylindrical sampling space. The algorithm is based on the improved RRT* algorithm of the traditional RRT algorithm. By constraining the generation space of the nodes, the generation path is in a very small cylindrical space. Compared with the traditional RRT algorithm, the path generated by the optimization algorithm is smooth and short, almost no redundant path, and is closer to the ideal route.

II. Traditional RRT algorithm AND RRT* algorithm

A. Traditional RRT algorithm

The extension process of the traditional RRT algorithm is shown in Fig. 1. In the case that the starting point is the root node, nodes are randomly sampled in the space, Then find a point in the RRT tree node that is closest to the random point, and the search direction is determined by these two nodes. Finally according to the search step, the new node is extended by the nearest point along the search direction.And then added it to the tree.After the above steps are completed, the loop search process is started. When the node of the random tree contains the target point or enters the target area, a path can be find in the random tree from the initial point to the target point [6].

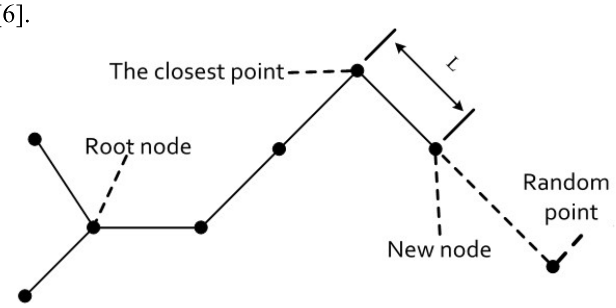


Fig.1 Extension process of traditional RRT algorithm

B. RRT algorithm*

In order to get a less expensive path and speed up the planning, this paper uses RRT* algorithm as the basic algorithm to optimize. Unlike traditional RRT, the RRT* algorithm adds branch trimming process. After generating the random point qrand, the algorithm finds the point qnearest (point2) closest to the qrand on the tree. Then, according to the

This work is partially supported by the Major Scientific and Technological Innovation Project of Shandong Province with Grant No. 2017CXGC0923, and Key Research and Development Program of Shandong Province Grant No. 2017GGX30112 and 2018GGX103025.

978-1-7281-6321-5/19//$31.00 ©2019 IEEE

step size, a point qnew is generated in advance, and then a circle with radius R is generated centered on qnew, and all points in the circle are found (each point has the corresponding cost, and the cost represents the distance to this point from the root node to the current point). If dist(point1,qnew)+cost(point1) < cost(point2)+ dist (point1, qnew), then the parent of qnew is point1, if dist (point2,qnew)+cost(qnew)< cost(point2), then select qnew as the parent node of point2. The pruning principle of the algorithm is shown in Fig. 2 [7].

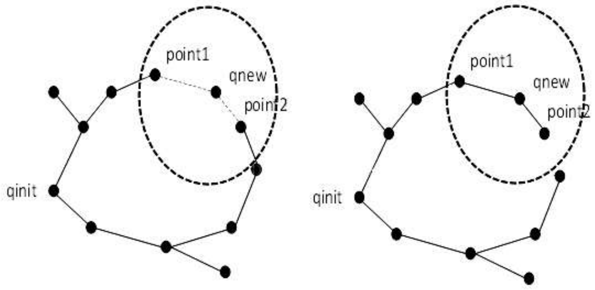


Fig.2 RRT* branch trimming principle

III. IMPROVED RRT ALGORITHM

A. *Improved motivation of traditional RRT algorithm*

Although the traditional RRT algorithm can generate the path to avoid obstacles, the generated path is often far from the ideal route due to the randomness of node selection. As shown by the black broken line in Fig. 3, this is a randomly generated path of the RRT algorithm under a given obstacle condition. It can be seen from the figure that the path generated by the conventional RRT algorithm has a path meaningless twist.

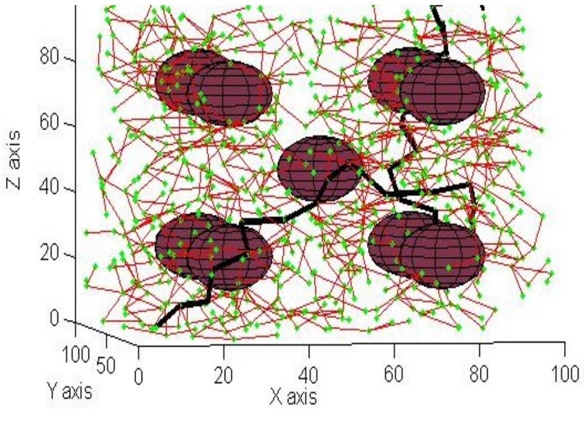


Fig.3 Two paths generated by the original RRT algorithm

B. *Ideas about the new RRT algorithm*

To solve this problem, it is necessary to adjust the sample space of the generated node. Fig. 4 shows the idea of path optimization in a two-dimensional space. The optimization idea is to form an ellipse with the starting point, the target point and the currently generated path, and regenerate the path as a new sample space. After that, the shape of the ellipse is adjusted, and as the ellipse continues to shrink, the path is continuously optimized. If there is no obstacle between the starting point and the target point, as the ellipse decreases, the final path will converge into a straight line, The process as shown in Fig. 5.

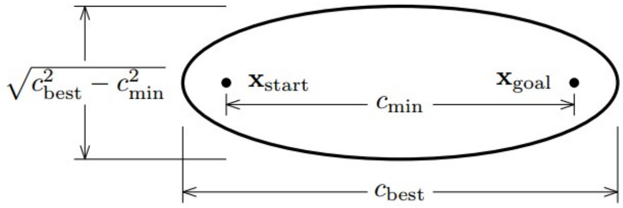


Fig.4 An ellipse consisting of the starting point and the target point

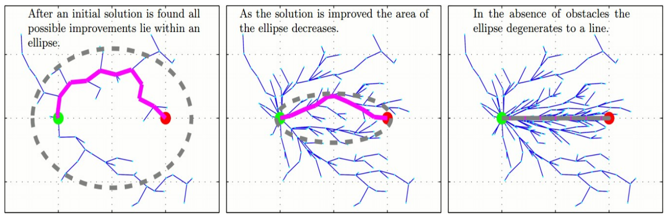


Fig.5 The path converges into a straight line

Drawing on this idea, for path generation in three-dimensional space, an ellipsoid similar to the above figure can be generated first, and then the ellipsoid is continuously reduced to achieve the purpose of path optimization. However, it is also very challenging to generate an ellipsoid in three-dimensional space and to make the nodes randomly and uniformly generated in this ellipsoid. In addition, this method will greatly increase the running time of the algorithm. This paper compares the regular geometry in a variety of three-dimensional spaces, and finally chooses to use the cylinder as the newly generated sample space.

The sample space with random sampling of the cylinder has two major advantages: First, it is easy and fast to generate the sample space, and the target point of the starting point is the center of the bottom surface, and the node with the farthest distance from the axis of the cylinder on the current path is selected as the radius of the cylinder. The cylinder is simple and has fewer parameters. Secondly, it is easier to randomly generate nodes in the cylinder. It can be determined by determining the distance between the newly generated nodes and the axis by using the starting point target point as the axis. Additional sample space changes are required, and only the judgment conditions need to be added when the node is generated.

C. *The process of generating node*

In Matlab, there is a known point A in three-dimensional space, and the distance formula of another straight line passing through two points of known B and C is: $d = norm(cross(C - B, P - B))$. Where A, B, and C are the coordinates of the point in space. The node is regarded as point A, and the starting point target point is regarded as two points of B and C. With this formula, the distance d of all nodes to the starting point point connection line can be calculated. Using the wire as the axis, d is the radius r of the bottom surface of the cylinder for each node. By taking the maximum value dmax of d, the new sample space with the largest radius as the generation node can be obtained, as shown in Fig. 6.

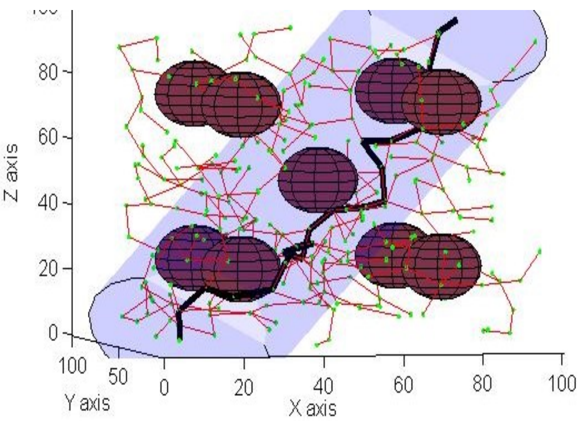


Fig.6 New sample space node generation algorithm

The new sample space does not need to be rebuilt. It only needs to add a decision condition when generating a new node: when the connection point of the generated node to the starting point is greater than or equal to dmax, it means that the node is not in the new sample space. Discard this node and regenerate the new node. Repeat the loop until the distance between the generated node and the connection is less than dmax, so that all new nodes are in this cylindrical sample space.

At this time, there is a feasible path in the cylinder with the starting point target point as the bottom center and the dmax as the bottom radius. If you want to optimize this path, you need to know if there is a feasible path in the cylinder with a smaller radius. Then you can change the sample space of the generated node to a cylinder with a smaller radius. In this paper, the reduction is set to 1/5 of the previous dmax. Run the algorithm again with the new cylinder as the sample space to see if a feasible path can be obtained. If the path is obtained, it indicates that the optimization is successful in this step. Continue to reduce the radius of the bottom surface of the cylinder and repeat the above cycle. If the path is not obtained, the optimization cannot be completed and the loop is ended.

The algorithm fake code of generating a new sample space node

```
function BuildNew_space(start_cord,end_cord,dmax )
 for i=1:dim
      randompoint ← (world.endcorner-world.origincorner)*rand
      new_point ← tree+(new_point/norm(new_point))*segmentlength;
      h ← start_cord,end_cord,new_point distance of the newly
generated node to the starting point target point
      while  h>=dmax
            new_point  --regenerate node
            h ← start_cord,end_cord,new_point-- ask for distance again
      if  h<dmax
            break
return New_space
```

D. Process of generating a path

Fig. 7 shows the path generated by the traditional RRT algorithm. The number of nodes required for this path generation is large and irregular, which greatly increases the running time of the program. Fig. 8 is the process of optimizing the algorithm to generate a path of given obstacle condition. Fig. 8(a) shows the path obtained by the program for the first time. The distance values of all nodes on the path to the cylinder axis are compared, and the value of dmax is taken as the cylinder radius of the new sample space. After that, the program enters the optimization loop, and the optimization process is terminated by continuously reducing the radius of the cylinder to reduce the sample space of the random node until the sample space is basically blocked by the obstacle and cannot generate a feasible path. As shown in Fig. 8(b), the path optimized by the RRT algorithm requires fewer nodes and the path is smoother and shorter than the path generated by the traditional RRT algorithm. Fig. 9 is the path generated by optimized RRT algorithm of random obstacle condition, the same conclusion as in Fig. 8 can also be drawn. The running time of the algorithm under given obstacles is 4.6s, under random obstacles is 5.1s. It can be seen that the proposed algorithm meets the requirements of rapid planning.

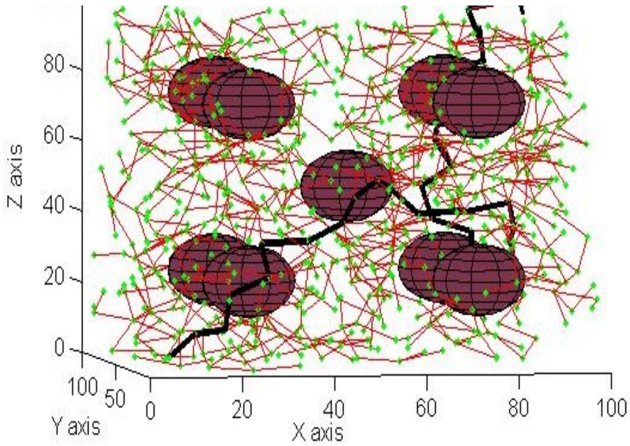
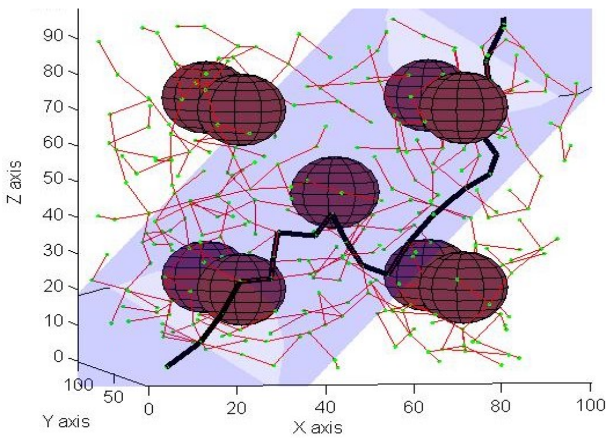
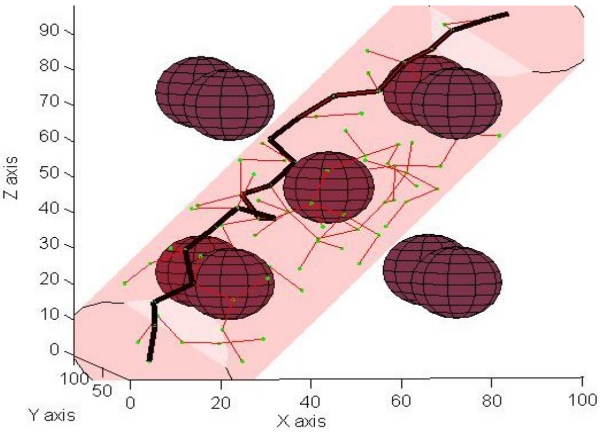


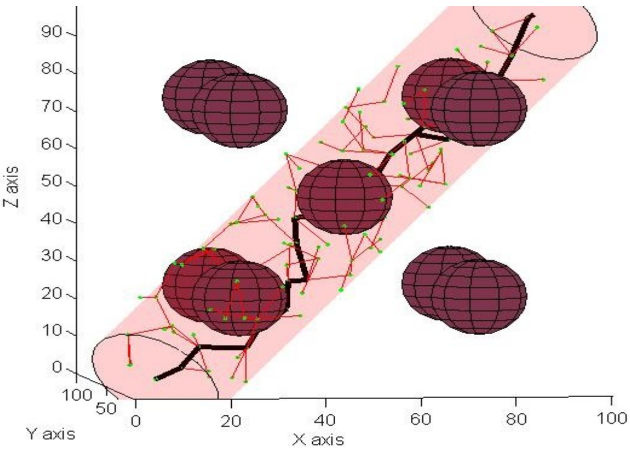
Fig.7 The path generated by the original RRT algorithm of given obstacle condition



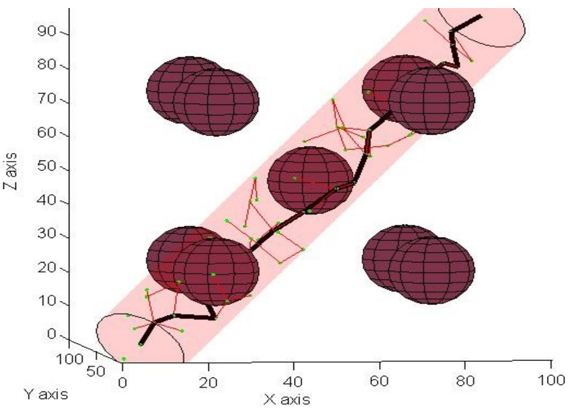
(a) The path for the first time of given obstacle condition



(b) The path for the second time of given obstacle condition

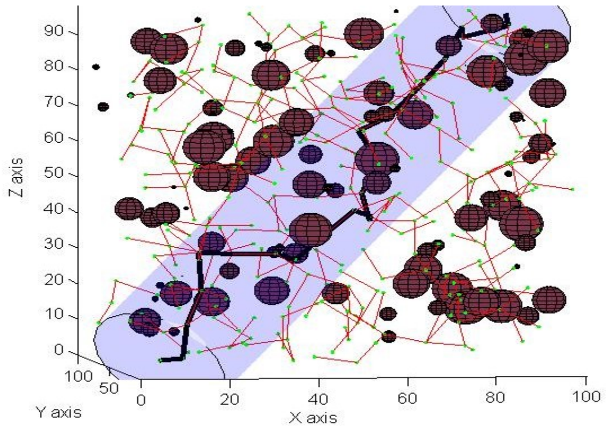


(c) The path for the third time of given obstacle condition

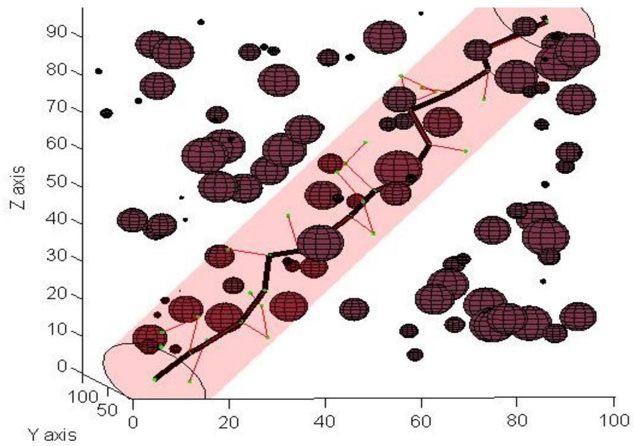


(d) The path for the last time of given obstacle condition

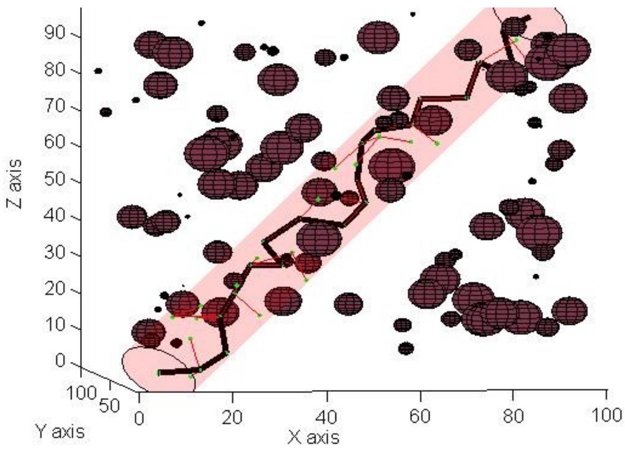
Fig.8 The path generated by optimized RRT algorithm of given obstacle condition



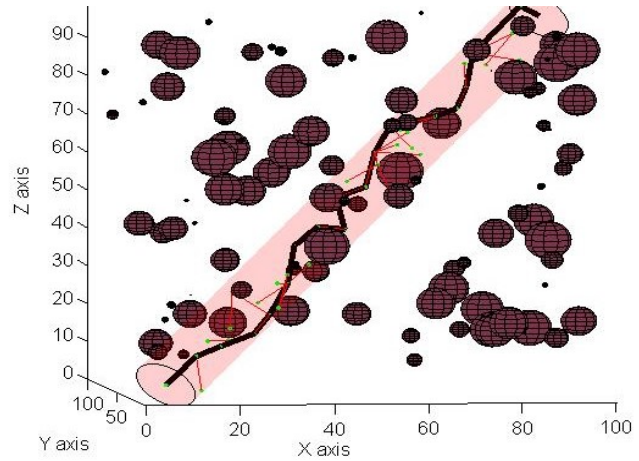
(a) The path for the first time of random obstacle condition



(b) The path for the second time of random obstacle condition



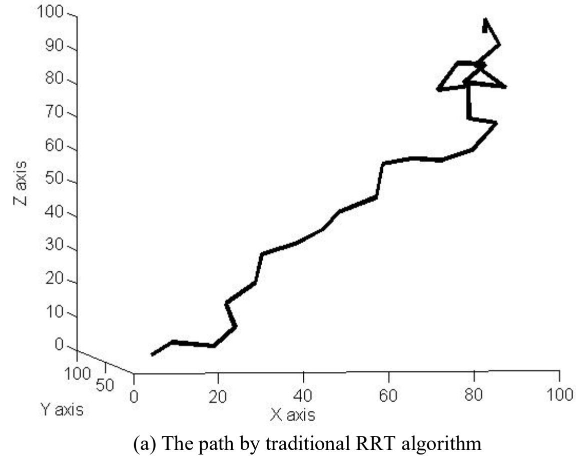
(d) The path for the third time of random obstacle condition



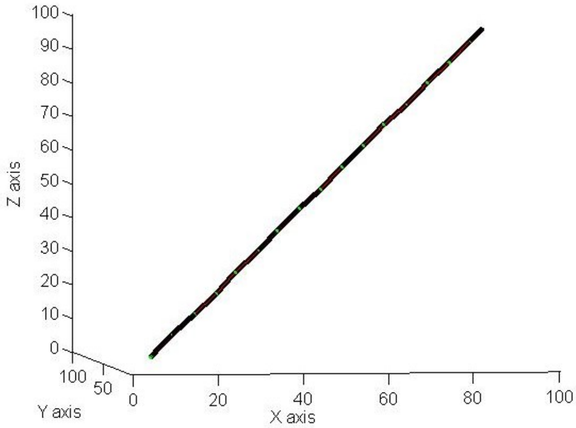
(d) The path for the last time of random obstacle condition

Fig. 9 The path generated by optimized RRT algorithm of random obstacle condition

Finally, the situation in which the surrounding environment does not have any obstacles is considered. The ideal route at this time is a straight line connecting the starting point to the target point. Fig. 10(a) is a path generated by the conventional RRT algorithm when the number of obstacles is set to zero. Obviously, the traditional RRT algorithm is difficult to generate the ideal path no matter how many times it is repeated. As shown in Fig. 10(b), the path generated by RRT optimization algorithm based on cylindrical sampling space. It can be seen from the figure that the path generated by the optimized RRT algorithm will converge to a straight line with the unrestricted reduction of the sample space due to the absence of obstacles, which is the same as our ideal route.



(a) The path by traditional RRT algorithm



(b) The path by optimized RRT algorithm

Fig. 10 Path by traditional and optimized RRT algorithm when no obstacles

IV. CONCLUSION

About the defect of traditional RRT algorithm in path planning, this paper proposes an RRT optimization algorithm based on cylindrical sampling space. The optimization algorithm uses the line connecting the starting point and the ending point of the path as the central axis. The maximum distance from all nodes on the path to the central axis is used as the bottom radius to generate a specific cylinder as the new node in the next path planning. It is ensured that there is a feasible path in the cylinder. On this basis, a new cylinder is generated by reducing the radius of the bottom surface as the current sample sampling space to continue to find the path. After that, repeat the above steps to achieve the purpose of path optimization. The path generated by the original RRT algorithm and the path generated by the optimization algorithm are compared and analyzed under the given obstacle and obstacle-free environment. The simulation results show that the path generated by the RRT optimization algorithm based on the cylindrical sampling space is smoother and shorter, almost no redundancy. The path meets the actual motion requirements of the robot. This paper proves the feasibility of the proposed algorithm only through the simulation environment of the spherical obstacles, but does not simulate the path planning under different types of obstacles. In future work, some simulation experiments will be added to verify the adaptability and robustness of this method through the path planning of different scenes. In order to make the path planning users can be more clear about the positioning and importance of the algorithm proposed in this paper, we will also add some RRT and other non modeling path planning algorithm comparison.

ACKNOWLEDGMENT

Grateful acknowledgements are given to the Major Scientific and Technological Innovation Project of Shandong

Province with Grant No. 2017CXGC0923, and Key Research and Development Program of Shandong Province Grant No. 2017GGX30112 and 2018GGX103025.

REFERENCES

[1] Daniel K, Nash A, Koenig S, "Theta any-angle path planning on grids," J. Artif Intell Res,2010,39:533-579.
[2] KHATIB O, "Real-time obstacle avoidance for manipulators and mobile robots," J. The International Journal of Robotics Research,1986,5(1):90-98.
[3] Liang JH, Lee CH, "Efficient collision-free path-planning of multiple mobile robots system using efficient artificial bee colony algorithms," J. Adv Eng Softw,2015,79:47-56.
[4] Noreen I, Khan A, Habib Z, "Optimal path planning using RRT* based approaches: a survey and future directions," J. Adv Comput Sci Appl (IJACSA) 2016,7:97-107.
[5] LAVALLES, "Rapidly-exploring random trees:a new tool for path planning" R. Computer Science Dept, lowa State University,1998.
[6] Karaman S, Frazzoli E, "ampling-based algorithms for optimal motion planning," J. Robot Res,2010,30(7):846-894.
[7] Qureshi AH, Ayaz Y, "Intelligent bidirectional rapidly-exploring random trees for optimal motion planning in complex cluttered environments," J. Robot Auton Syst,2015, 68:1-11.