

Improving Lateral Safety Distance-Based on Feature Detection and Probabilistic Roadmaps for Unmanned Vehicle Path Planning

Jianfeng Wang

Harbin Institute of Technology

Guangliang Chang (✉ changgl0404@163.com)

Harbin Institute of Technology <https://orcid.org/0000-0003-4485-0966>

Weihua Li

Harbin Institute of Technology

Na Yang

Harbin Institute of Technology

Boqian Wang

Harbin Institute of Technology

Yuhui Sun

Harbin Institute of Technology

Research

Keywords: path planning, path points selection, collision-free, feature detection

Posted Date: August 13th, 2020

DOI: <https://doi.org/10.21203/rs.3.rs-54747/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

1 Improving Lateral Safety Distance-Based on Feature 2 Detection and Probabilistic Roadmaps for Unmanned 3 Vehicle Path Planning

4 Wang Jianfeng¹, Chang Guangliang^{1,*}, Weihua Li¹, Na Yang¹, Wang Boqian¹, Sun Yuhui¹

6 ¹School of Automotive Engineering, Harbin Institute of Technology, Weihai 264209, China

7 *Corresponding author E-mail: changgl0404@163.com

8 Abstract

9 Most of the existing path-planning algorithms do not consider lateral safe distance requirements in
10 practical applications. Hence, in this study, a new path point selection algorithm is proposed for path
11 planning. The algorithm first used the Harris and Line Segment Detector(LSD) algorithms to detect and
12 obtain the corner and edge information of obstacles. A vertical line was provided to the edge of the
13 surrounding obstacles along each corner successively. In this process, the narrow impassable area in the
14 map was filtered and removed by setting a safety threshold, and the foot of the vertical coordinates were
15 simultaneously obtained. The corresponding midpoint coordinates were solved by using the corner
16 coordinates and the foot of the corresponding perpendicular coordinates. The midpoint coordinates were
17 used as candidate points to generate path points. These candidate points are screened and relaxed using
18 the Probabilistic Roadmaps(PRM) algorithm to obtain the series of path points required. Finally, the path
19 was planned according to these path points and smoothed using the Quadratic polynomial interpolation
20 method(QPMI). Through simulation experiments, the method proposed in this study can solve a unique
21 path without randomness under given conditions, and the probability of a collision in practical applications
22 was reduced.

23

24 **Keywords** path planning, path points selection, collision-free, feature detection

1 1. Introduction

2 Path planning, as one of the key technologies for driverless driving has been a hot research topic, attracting
3 the attention of a large number of researchers. It is designed to enable an autonomous vehicle to achieve
4 a series of collision-free and safety motions in a given environment to accomplish certain tasks. Traditional
5 path searching and planning algorithms have many types, such as rapidly exploring random tree path
6 planning, sub-goal networks, A* algorithms and D* algorithms, artificial potential fields, particle swarm
7 optimization (PSO) algorithms, and polynomial interpolation methods [1–4]. With the increasing
8 complexity of the environment and the difficulty of tasks, these traditional methods are, however, often
9 incapable of achieving the ultimate performance.

10 In [5], the cubic polynomial model was used to solve the constraint conditions and objective functions,
11 and a dynamic optimal trajectory model was obtained. However, this study was based on a relatively
12 representative scenario, which investigated local path planning, and only considered the longitudinal safe
13 distance, while other more complex road conditions were not completely considered. In [6–8], the artificial
14 potential energy field model was used to plan the road, in consideration of a dynamic target position and
15 road boundary, and simulations were carried out, the results of which were deemed to be satisfactory.
16 However, the artificial potential energy field model cannot address the problems of local minima, and the
17 target cannot be satisfactorily reached. The researchers in [9] proposed an improved artificial potential
18 field based on the regression search method for mobile robot path planning, which can plan a short solution
19 from the robot's position to a target position. This method solved the problems of local minimization and
20 oscillatory motion and used the regression search (RS) method to optimize the planning path. However,
21 the smoothness of the path planned using this method cannot meet the driving requirements of autonomous

1 vehicles. Li et al. [10] selected the cubic spline method to generate multiple possible paths and proposed
2 an improved PSO algorithm to obtain the optimal path. However, this method was not suitable for more
3 complex environments and cannot handle irregular obstacles. You and Li et al. [11] proposed a new local
4 path planning approach based on optimization methods with probabilistic completeness. They introduced
5 Hamiltonian Monte Carlo algorithm to their modification, which constantly forced the initial path to jump
6 out of the local extremum, thus improving the robustness and success rate of their path planning approach.
7 However, this method currently cannot sufficiently address the path planning problems of large maps and
8 maze-like maps. The researchers in [12] used the ant colony algorithm to address the problem of unmanned
9 vehicle path planning. Liang et al. [13] proposed control strategies for the torque of each wheel and the
10 rear-wheel-steering angle to maintain a stable velocity by using second-order sliding mode (SOSM)
11 techniques. Tan et al. [14] proposed a novel method of global optimal path planning for mobile robots
12 based on the improved Dijkstra algorithm and ant system algorithm. Li et al. [15] proposed methods based
13 on kinematics and dynamic constraints to solve path planning problems for unmanned vehicles. Yun et al.
14 [16] suggested a path-planning algorithm that considered the stability of unmanned ground vehicles based
15 on the genetic algorithm. Cai et al. [17] introduced the PSO algorithm as one of the new swarm intelligent
16 optimization methods into a path planning for autonomous vehicles, which comprised particle
17 representation methods for vehicle routing problems with fast convergence speeds. Fethi, D et al. [18]
18 proposed a new approach of optimal simultaneous localization, mapping, and path planning based on the
19 optimal control theory and simultaneous localization and mapping. As an initial idea for solving a min-
20 max multiple-depot heterogeneous traveling salesman problem, Bae, J. et al. [19] proposed new heuristics
21 for the path planning problem of two heterogeneous unmanned vehicles. To plan a global path minimizing

1 the risk of an unmanned vehicle on the battlefield, Shin, J. et al. [20] proposed a global path replanning
2 method. Arokiasami, W. A. et al. [21] proposed the Service Oriented Interoperable Framework for Robot
3 Autonomy (SOIFRA), which provided collision avoidance, path planning, and tracking behaviors for
4 unmanned ground vehicles. To find feasible solutions and lower bounds for the path planning problem,
5 Sundar, K. et al. [22] developed a primal-dual heuristic and incorporated the heuristic into a Lagrangian
6 relaxation procedure.

7 In path planning, most of the above algorithms simplify the control vehicle and do not completely
8 consider its actual body size. When the actual control vehicle is driven, deviations always exist in the
9 control technology, making the vehicle likely to collide as its lateral safe distance is insufficient when it
10 actually follows the planned path. To address this problem, a new path point selection method for path
11 planning is proposed in this study. After a map is obtained, the algorithm is used to detect the corner and
12 edge information of each obstacle successively. Then, it takes each corner as a starting point to make a
13 vertical line to the surrounding obstacles, calculating the corresponding vertical foot coordinates. In this
14 process, the narrow impassable areas in the map are filtered out by setting a security threshold.
15 Subsequently, the corresponding midpoints are obtained by calculating the corners and vertical foot
16 coordinates, and these midpoints are regarded as the path points for path planning. Finally, these path
17 points are used to plan the path and smoothen it. The path planned using this method can ensure that the
18 path is far removed from obstacles on both sides, thereby maximizing the safety of the driven vehicle.

19 The remainder of this paper is organized as follows. Section 2 describes in detail the algorithm proposed
20 in this study and the problems we intend to address. Section 3 discusses the implementation steps and
21 solutions of the algorithm proposed in this study. Section 4 presents the simulations performed on the

1 proposed algorithm, the results of which are analyzed. Section 5 concludes this paper.

2

3 **2. Problem Description**

4 In existing path planning algorithms, most of them simplify the control vehicle itself. Although the path
5 generated by existing algorithms can meet the requirements of G2 continuity and smoothness, it ignores
6 the size of the vehicle itself. Consequently, the planned path is considerably close to obstacles in some
7 locations, increasing the risk of a collision in practical applications.

8 Considering these problems, we propose a new path planning algorithm in this study, which aims to
9 eliminate the possibility of vehicle collisions in the process of path planning. In existing path planning
10 algorithms, most of the collisions are caused by insufficient lateral distance. Fig. 1 shows several cases of
11 path point selection—the red area representing the obstacle and the white area representing the passable
12 area. Provided that the planned path can pass through the blue triangular points in the figure, the control
13 vehicle can maximize its distance from obstacles on both sides of the path simultaneously. The main
14 objective of our work on this algorithm is to extract the blue triangular points for path planning.

15

16

Fig. 1 Several cases of path point selection

17 The implementation steps of the proposed algorithm are as follows:

18 a) Detection to obtain the corner coordinates of the obstacle in the full map

19 b) Detection to obtain the edge information of the obstacle in the whole image

20 c) Creation of a vertical line along the corner to the edge of the obstacle to obtain the coordinates of the

1 foot of perpendicular

2 d) Obtaining of the corresponding midpoint coordinates according to the corner coordinates and the
3 corresponding vertical coordinates

4 e) Filtering of the obtained midpoint coordinates and the generation of the initial path, and

5 f) Smoothing the path.

6

7 **3. Methods**

8 **3.1 Detection of the corner of the obstacle**

9 At present, corner detection algorithms can be divided into three categories: corner detection based on a
10 grayscale image, corner detection based on a binary image, and corner detection based on a contour curve.
11 Corner detection algorithms based on a grayscale image include the gradient, template, and template
12 gradient methods. In the template-based method, it considers the gray change of a pixel region and
13 compares it with the brightness of its neighborhood, the large point of the neighborhood being defined as
14 the corner. The commonly used template-based corner detection algorithms include the Kitchen-Rosenfeld
15 corner detection algorithm, Harris corner detection algorithm, KLT algorithm, and SUSAN corner
16 detection algorithm[23]. In this study, we selected the Harris corner detection algorithm to address our
17 corner detection problems and obtain corner coordinates.

18 **3.2 Detection of the edges of obstacles**

19 In this study, the map for path planning was simplified, the edges of obstacles all being simplified to
20 straight lines. Therefore, when detecting the edge of an obstacle, the problem was the detection of a

1 straight-line segment on the map. To address this problem, the LSD algorithm [24] was used. The LSD
 2 algorithm is a linear timeline detector, which can provide accurate sub-pixel results. It is designed to work
 3 on any digital image without the adjustment of parameters. It controls the number of false positives: on
 4 average, one false alarm is allowed per image. After algorithm detection, the edge information of the
 5 obstacle was obtained, as shown in Table 1.

6 Table 1
 7 Edge information of obstacles

Slope	Abscissa of end point 1	Ordinate of end point 1	Abscissa of end point 2	Ordinate of end point 2	Edge length
k	x_1	y_1	x_2	y_2	len

8 3.3 Creation of vertical lines from the corner to the edge of obstacles

9 The required corner coordinates and obstacle edge information were detected in the first two steps.
 10 Thereafter, each corner was used as a starting point, and vertical lines were made to all of the detected
 11 edges of the obstacle successively—the corresponding coordinates of the foot of perpendicular were
 12 obtained.

13 To obtain the foot of perpendicular, the solution was divided into three cases. In the first case, when
 14 $|k| \geq 100$, let $k = \text{Inf}$, and the foot of perpendicular was solved using equation 1:

$$\begin{aligned}
 x_p &= x_1 \\
 y_p &= y_c
 \end{aligned}
 \tag{1}$$

16 where x_p represents the horizontal coordinate of the foot of perpendicular, y_p represents the
 17 longitudinal coordinate of the foot of perpendicular, and y_c represents the longitudinal coordinate of the
 18 corner point.

19 In the second case, when $|k| \leq 0.01$, let $k = 0$, and the foot of perpendicular was solved using
 20 equation 2:

1
$$\begin{aligned} x_p &= x_c \\ y_p &= y_1 \end{aligned} \quad (2)$$

2 where x_c represents the horizontal coordinate of the corner point.

3 In the third case, when $0.01 < |k| < 100$, the foot of perpendicular was solved using equation 3:

4
$$\begin{aligned} k_p &= -\frac{1}{k} \\ \begin{bmatrix} x_p \\ y_p \end{bmatrix} &= \begin{bmatrix} -k & 1 \\ -k_p & 1 \end{bmatrix}^{-1} \begin{bmatrix} y_1 - kx_1 \\ y_c - k_px_c \end{bmatrix} \end{aligned} \quad (3)$$

5 where k_p represents the slope of the vertical line at the edge of the obstacle.

6 In this step, in order to achieve collision-free operation as much as possible in the process of path
7 planning, a threshold was set for the distance from the corner to the edge of the obstacle. As the actual
8 vehicle requires sufficient lateral distance when driving and some deviations exist during tracking control
9 process, the distance from the corner to the edge of the obstacle is required to be more than 6 m. Thus,
10 some impassable areas can be filtered out during the path planning process. The distance is solved using
11 equation 4:

12
$$d = \sqrt{(x_c - x_p)^2 + (y_c - y_p)^2} \quad (4)$$

13 where d represents the distance from the corner to the edge of the obstacle.

14 **3.4 Solving the midpoint coordinates**

15 In the previous step, some impassable areas are screened out by setting the threshold of the distance from
16 the corner to the obstacle edge. However, in the process of path planning, if the path point used to generate
17 the path is not suitable, it may still lead to a path that is too close to an obstacle. This increases the
18 possibility of a collision in practical application, making the selection of path points extremely important.
19 Through analysis, if the planned path passes through the midpoint between obstacles, it can meet safety
20 requirements—the midpoint between the above-mentioned corner point and the corresponding foot of

1 perpendicular achieves this. Therefore, through the calculation of the corner coordinates and the
2 corresponding vertical coordinates, the midpoint coordinates for path planning were obtained, which can
3 be solved using equation 5:

$$4 \quad \begin{bmatrix} x_m \\ y_m \end{bmatrix} = \frac{1}{2} \left(\begin{bmatrix} x_c \\ y_c \end{bmatrix} + \begin{bmatrix} x_p \\ y_p \end{bmatrix} \right) \quad (5)$$

5 where x_m represents the horizontal coordinate of the midpoint and y_m represents the longitudinal
6 coordinate of the midpoint.

7 **3.5 Generating the initial path**

8 As the obtained midpoint coordinates are all over the map and some of the previous steps do not filter the
9 data, some midpoints will fall into obstacle areas, so they cannot be used directly in the path planning
10 process and need to be further screened.

11 In this study, we used the PRM algorithm [25] to process the data obtained in the previous step. The
12 PRM algorithm automatically filters and deletes midpoints that fall in the obstacle area. Meanwhile,
13 through a repeated relaxation process, the PRM algorithm obtains the midpoint required for the shortest
14 path from the starting point to the end point—that is, a path point that can be used to generate the path
15 itself.

16 **3.6 Smoothing the path**

17 Given that the path generated by the PRM algorithm is piecewise linear and its smoothness and continuity
18 do not meet the requirements of driverless vehicles, the path should be smoothed. We used the QPMI
19 method [26] to smooth the path, which combines the fuzzy control membership function with the
20 polynomial interpolation method to realize path smoothing. A path processed using this method meets the

1 requirements of practical applications.

2

3 **4. Simulation and Analysis**

4 According to the implementation steps of the above algorithm, we wrote a program to solve the problem.

5 The corner detection results are shown in Fig. 2. The green dots in the picture represent the obstacle
6 corners detected by the Harris algorithm. The figure shows that the detection results are accurate, with no
7 omissions or error detection.

8

9

Fig 2. Corner detection results

10 Fig. 3 shows the vertical foot calculated using the algorithm—that is, the purple square points in the
11 figure. The figure show that the calculated results of the detection are precise, without omission or error
12 detection. Further, some of the foot of perpendicular points are close to each other or even overlap, which
13 has a minimal impact on follow-up calculations. Through the threshold set, the feet of perpendicular close
14 to corresponding corners have been filtered out, and the rest of the feet of perpendicular belong to different
15 corners.

16

17

Fig 3. Results of the foot of perpendicular

18 The midpoint coordinates calculated by the algorithm are the blue triangular points shown in Fig. 4.
19 The figure shows that the midpoint coordinate calculation results are consistent with expectations and all
20 of the midpoints of the nearest distance from the corner of the obstacle to the edge of the surrounding

1 obstacle are correctly calculated. These midpoints are far removed from obstacles on both sides, which
2 meets the requirements of practical applications.

3

4

Fig 4. Midpoint detection results

5 The processing result of the ordinary PRM algorithm is shown in Fig. 5. It obtains a linear path by
6 randomly placing points on the entire graph. Using this method of randomly scattered points, if the number
7 of sampling points is insufficient, it leads to no path being generated. Even if the number of sampling
8 points is adequate, to create the shortest path in the process of generating a path, the final path generated
9 is too close to obstacles in some places, as shown in Fig. 5. This makes the application of the path more
10 difficult. In addition, this randomness renders the sampling points generated by different operations
11 different, and the path generated by these path points is not unique. This leads to an increased risk of
12 collision in the process of subsequent smoothing and practical application. Further, the randomness of the
13 algorithm itself makes it more difficult to address this problem.

14

15

Fig 5. Ordinary PRM algorithm

16 Our algorithm uses the previously calculated midpoint for the PRM algorithm, as shown in Fig. 6. The
17 figure shows that the distance between the path and the obstacle calculated by the midpoint coordinates is
18 farther and safer than that of the ordinary PRM algorithm. Moreover, since the position of the midpoint
19 does not change, the result of each calculation is unique, which remarkably improves the application and
20 controllability of the resulting path.

21

1 **Fig 6.** PRM algorithm using midpoint coordinates

2 The result of path smoothing using the QPMI method is shown in Fig. 7—the orange curve in the figure.
3 The path shown in Fig. 7(a) is based on the ordinary PRM algorithm. The path generated using this method
4 is evidently too close to the obstacle in some places, and if this path were to be applied in practice, a
5 collision would be likely to occur. Fig. 7(b) shows the path planned using the proposed algorithm. This
6 path meets the requirements of smoothness and continuity of driverless vehicle driving paths. Compared
7 to Fig. 7(a), the path generated using the proposed method is safer and farther away from all obstacles,
8 which meets the safety requirements for practical applications, reducing the risk of collisions. Meanwhile,
9 since the path point used for path planning is unique, the path generated using this method is also unique,
10 which renders the path easier to control in practical applications.

11 During simulation, the proposed method was used to improve the selection of path points for path
12 planning. Through simulation and analysis, we found that this method clearly improves the path generated
13 by the algorithm. The planned path can meet the requirements for a driverless vehicle driving path while
14 remarkably reducing the risk of collision in practical applications.

15

16 **Fig 7.** Smooth path with QPMI algorithm. a) Ordinary QPMI b) QPMI based on the proposed method

17

18 **5. Conclusions and Discussion**

19 In existing path planning methods, only few considers the problem of lateral safe distance in the driving
20 process, thereby increasing the risk of collision. In this study, we proposed a method to improve the

1 selection of path points. Our method successively used the Harris and LSD algorithms to detect the corner
2 and edge information of obstacles in a given map. Using the corner as a starting point, a vertical line was
3 provided to the edge of the surrounding obstacles successively, and the corresponding vertical coordinates
4 were obtained. In the process of establishing the vertical foot, the narrow impassable areas in the map
5 were removed by setting a safety threshold. The corresponding midpoint coordinates were obtained using
6 the corner points and the corresponding perpendicular coordinates, which were used as the candidate
7 points for each path point needed in the path planning process. After the starting and end points of a path
8 were given, the candidate points were screened using the PRM algorithm to determine a path of the
9 shortest length. Finally, the QPMI method was used to smooth the path.

10 In this study, simulation experiments were carried out on the proposed algorithm and compared to the
11 ordinary PRM algorithm. The results showed that, after selecting the starting and end points, the proposed
12 method could calculate a unique path, which met the requirements of smoothness and continuity of a
13 driverless vehicles path and did not exhibit the randomness of the path generated using the ordinary PRM
14 algorithm. Further, the path generated using this method remarkably reduced the risk of collision in
15 practical applications, improving the safety and controllability of autonomous driving in practical
16 applications.

17 The proposed algorithm simplified the map and the obstacles into graphics with clear corners and
18 straight edges. Therefore, this method is currently only suitable for maps with clear corner points and
19 straight edges, but not for maps with irregular edges. Our follow-up work will focus on the application of
20 this method to the path planning of irregular maps and further broaden the scope of availability of this
21 method.

1

2 **List of Abbreviations**

LSD	Line Segment Detector
PRM	Probabilistic Roadmaps
QPMI	Quadratic polynomial interpolation
PSO	Particle swarm optimization
SOIFRA	Service Oriented Interoperable Framework for Robot Autonomy
RS	Regression search
SOSM	Second-order sliding mode

3

4 **Declarations**

5 **Availability of data and materials**

6 The datasets used and/or analysed during the current study are available from the corresponding author on
7 reasonable request.

8 **Competing interests**

9 The authors declare that they have no competing interests

10 **Funding**

11 Not applicable

12 **Authors' contributions**

13 Chang G.L. proposed the idea of this article, explained the principle of the algorithm, and was a major
14 contributor in writing the manuscript. All authors read and approved the final manuscript.

15 **Acknowledgements**

16 Not applicable

1

2 References

- 3 [1] Wang W, Zuo L and Xu X. A learning-based multi-RRT approach for robot path planning in narrow
4 passages. *Journal of Intelligent & Robotic Systems* 2018; 90: 81–100.
- 5 [2] Luo, C., Zhu, A., Mo, H., Zhao, W., & Ieee. Planning Optimal Trajectory for Histogram-Enabled
6 Mapping And Navigation by an Efficient PSO Algorithm.In: *12th World Congress on Intelligent
7 Control and Automation (WCICA)*, Guilin, China, 12-15 June, 2016.
- 8 [3] A. Stentz (1995) The focussed D* algorithm for realtime replanning. In: *Proceedings of the
9 International Joint Conference on Artificial Intelligence (IJCAI)*, Lawrence Erlbaum Associates Ltd,
10 pp. 1652-1659.
- 11 [4] Kim, D. H. (2009). Escaping route method for a trap situation in local path planning. *International
12 Journal of Control, Automation and Systems*, 7(3), 495-500.
- 13 [5] Liu, Z.-q, Zhang, T., & Wang, Y.-f. Research on Local Dynamic Path Planning Method for Intelligent
14 Vehicle Lane-Changing. *Journal of Advanced Transportation*, 2019.
- 15 [6] J. Hu, L. Kong, W. Shu et al., “Scheduling of connected autonomous vehicles on highway lanes,”In:
16 *Proceedings of the 2012 IEEE Global Communications Conference*, pp. 5556–5561, IEEE, Anaheim,
17 Calif, USA, 2012.
- 18 [7] X. Huang, Research on Vehicle Moving Status Recognition Based on Vehicle Networking, South
19 China University of Technology, 2012.
- 20 [8] A. Pandey, “Mobile Robot Navigation and Obstacle Avoidance Techniques: A Review,” *International*

- 1 *Robotics & Automation Journal*, vol. 2, no. 3, pp. 1–12, 2017.
- 2 [9] Li, G., Yamashita, A., Asama, H., & Tamura, Y. An efficient improved artificial potential field based
3 regression search method for robot path planning. *IEEE International Conference on Mechatronics
4 and Automation*, 2012.
- 5 [10] Li, W., Tan, M., Wang, L., & Wang, Q. A cubic spline method combining improved particle swarm
6 optimization for robot path planning in dynamic uncertain environment. *International Journal of
7 Advanced Robotic Systems*, 17(1), 2020.
- 8 [11] You, B., Li, Z., Ding, L., Gao, H., & Xu, J. (2019). A new local path planning approach based on
9 improved dual covariant Hamiltonian optimization for motion planning method. *Advances in
10 Mechanical Engineering*, 11(5).
- 11 [12] Yue, L., & Chen, H. (2019). Unmanned vehicle path planning using a novel ant colony algorithm.
12 *EURASIP Journal on Wireless Communications and Networking*, 2019(1).
- 13 [13] Zhongchao Liang, Jing Zhao, Zhen Dong, Yongfu Wang, and Zhengtao Ding. Torque Vectoring and
14 Rear-Wheel-Steering Control for Vehicle's Uncertain Slips on Soft and Slope Terrain Using Sliding
15 Mode Algorithm. *IEEE Transactions on Vehicular Technology*, 2020, 69(4): 3805-3815.
- 16 [14] Tan, G. Z., He, H., & Sloman, A. (2006). Global optimal path planning for mobile robot based on
17 improved Dijkstra algorithm and ant system algorithm. *Journal of Central South University of
18 Technology*, 13(1), 80-86.
- 19 [15] Li, L., Zhong, B., & Geng, Z. (2017). Study on Path Planning of Unmanned Vehicle Based on
20 Kinematic and Dynamic Constraints. In D. Yue, C. Peng, D. Du, T. Zhang, M. Zheng, & Q. Han (Eds.),
21 *Intelligent Computing, Networked Control, and Their Engineering Applications*, Pt Ii (Vol. 762, pp.

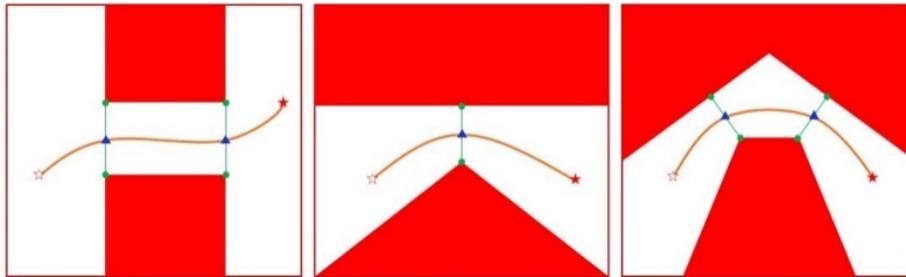
- 1 94-104).
- 2 [16] Yun, S., & Won, M. (2017). Genetic Algorithm Based 3D Environment Local Path Planning for
3 Autonomous Driving of Unmanned Vehicles in Rough Terrain. *Journal of the Korea Institute of*
4 *Military Science and Technology*, 20(6), 803-812.
- 5 [17] Cai, L., Jia, J., & Lei, J. (2015). Research on Path Optimization with PSO for Unmanned Vehicle.
6 *International Journal of Online Engineering*, 11(8), 21-24.
- 7 [18] Fethi, D., Nemra, A., Louadj, K., & Hamerlain, M. (2018). Simultaneous localization, mapping, and
8 path planning for unmanned vehicle using optimal control. *Advances in Mechanical Engineering*,
9 10(1).
- 10 [19] Bae, J., & Chung, W. (2019). Heuristics for Two Depot Heterogeneous Unmanned Vehicle Path
11 Planning to Minimize Maximum Travel Cost. *Sensors*, 19(11).
- 12 [20] Shin, J., Kwak D. J. (2016). An Approach to Global Path Replanning Method Considering 4D
13 Environmental Information. *Journal of the Korea Institute of Military Science and Technology*, 19(6),
14 779-788.
- 15 [21] Arokiasami, W. A., Vadakkepat, P., Tan, K. C., & Srinivasan, D. (2018). Real-Time Path-Generation
16 and Path-Following Using an Interoperable Multi-Agent Framework. *Unmanned Systems*, 6(4), 231-
17 250.
- 18 [22] Sundar, K., & Rathinam, S. (2013). A Primal-Dual Heuristic for a Heterogeneous Unmanned Vehicle
19 Path Planning Problem Regular Paper. *International Journal of Advanced Robotic Systems*, 10.
- 20 [23] Wang, Z., Li, R., Shao, Z., Ma, M., Liang, J., Liu, W., . . . Liu, Y. (2017). Adaptive Harris corner
21 detection algorithm based on iterative threshold. *Modern Physics Letters B*, 31(15).

1 [24] Grompone von Gioi, R., Jakubowicz, J., Morel, J.-M., & Randall, G. (2012). LSD: a Line Segment
2 Detector. *Image Processing On Line*, 2, 35-55.

3 [25] H. Choset, K.M. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L.E. Kavraki, S. Thrun (2005)
4 Principles of robot motion: theory, algorithms, and implementations, pp.107-262, Cambridge, MA:
5 MIT Press.

6 [26] Huh, U.-Y., & Chang, S.-R. (2014). A G2 Continuous Path-smoothing Algorithm Using Modified
7 Quadratic Polynomial Interpolation. *International Journal of Advanced Robotic Systems*, 11(2).

8

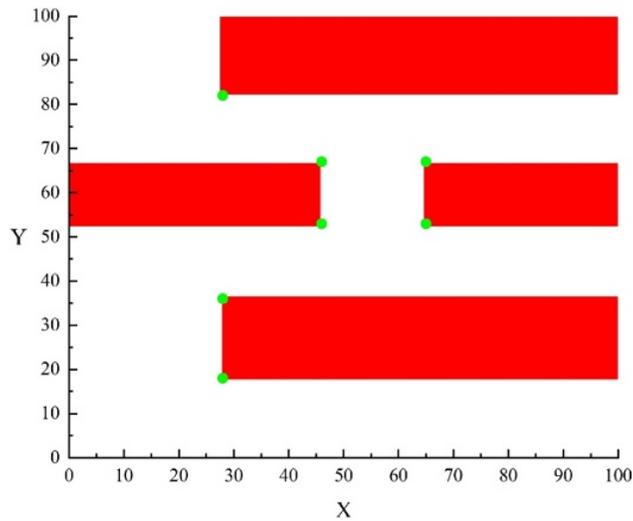


9

10

11

Fig. 1 Several cases of path point selection



12

13

14

Fig. 2 Corner detection results

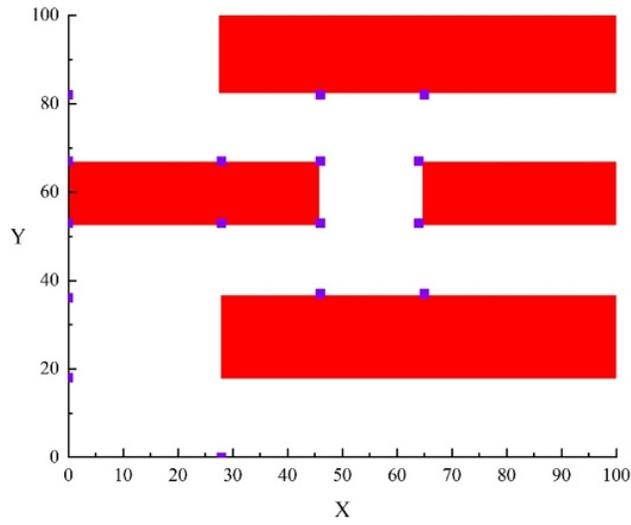


Fig. 3 Results of the foot of perpendicular

1
2
3

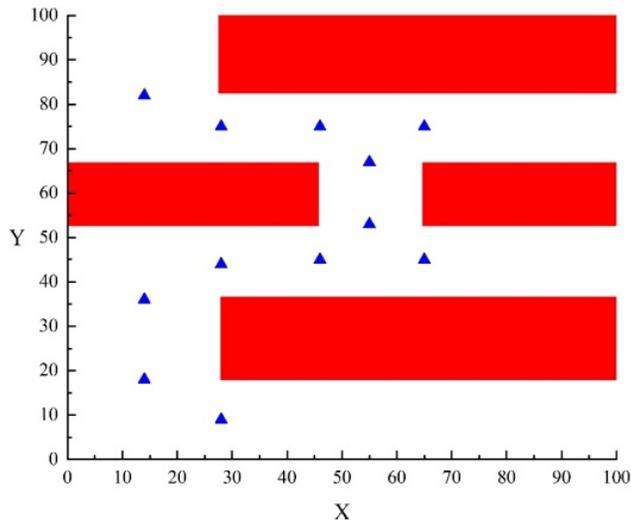


Fig. 4 Midpoint detection results

4
5
6

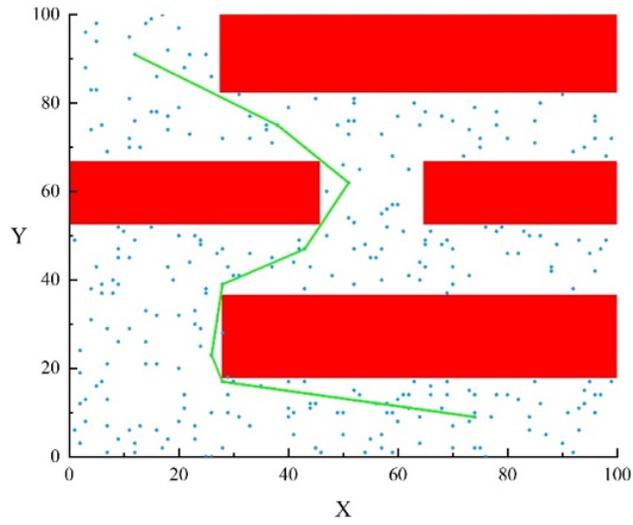


Fig. 5 Ordinary PRM algorithm

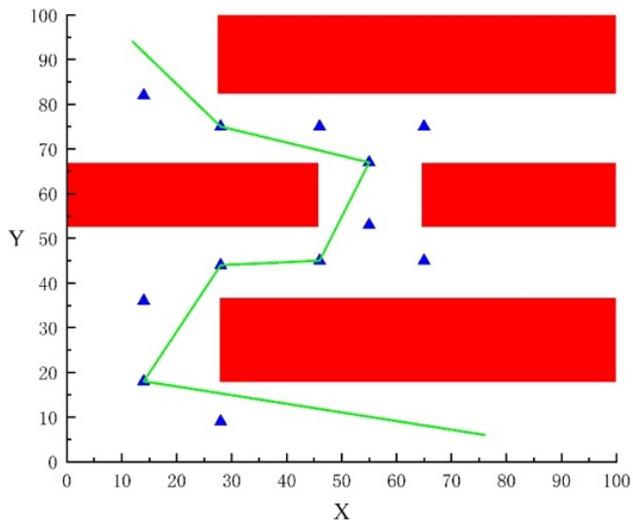


Fig. 6 PRM algorithm using midpoint coordinates

1

2

3

4

5

6

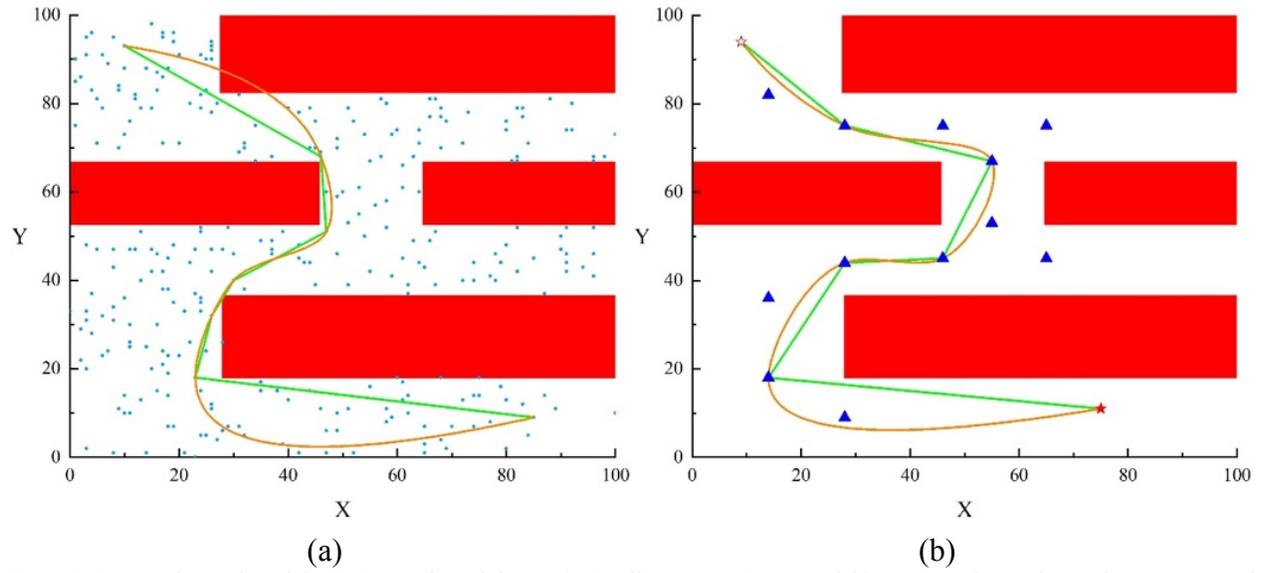


Fig. 7 Smooth path with QPMI algorithm. a) Ordinary QPMI and b) QPMI based on the proposed method

1
2
3
4
5

Figures

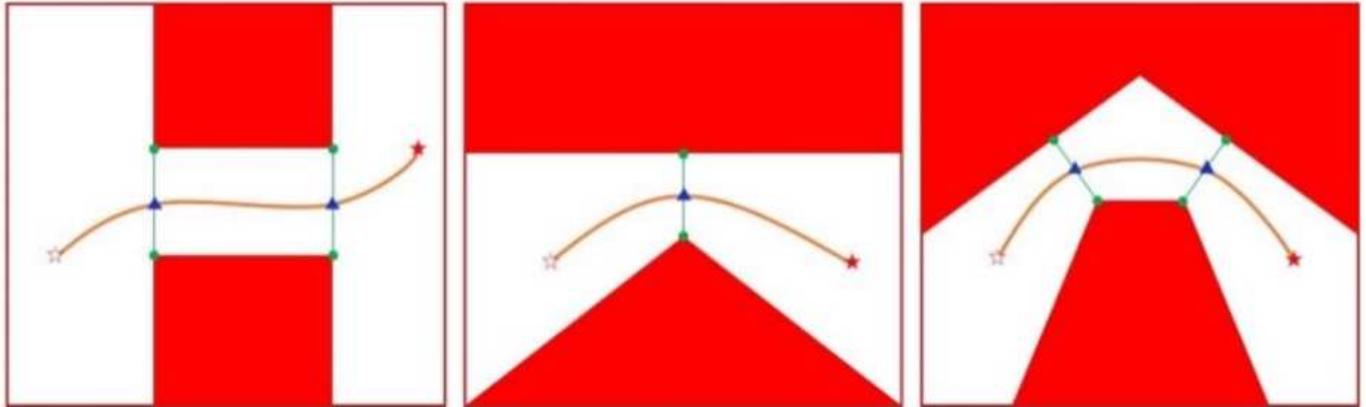


Figure 1

Several cases of path point selection

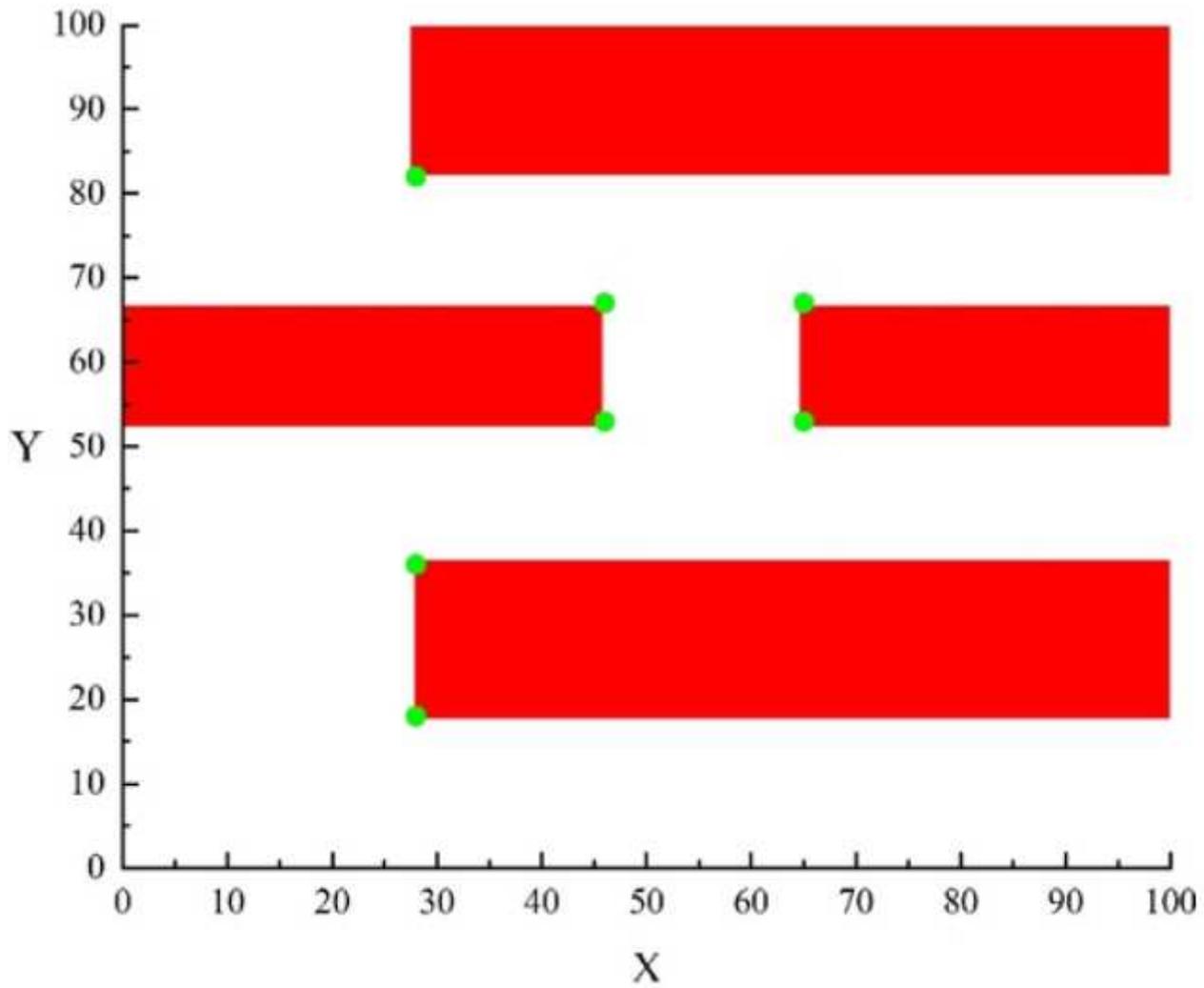


Figure 2

Corner detection results

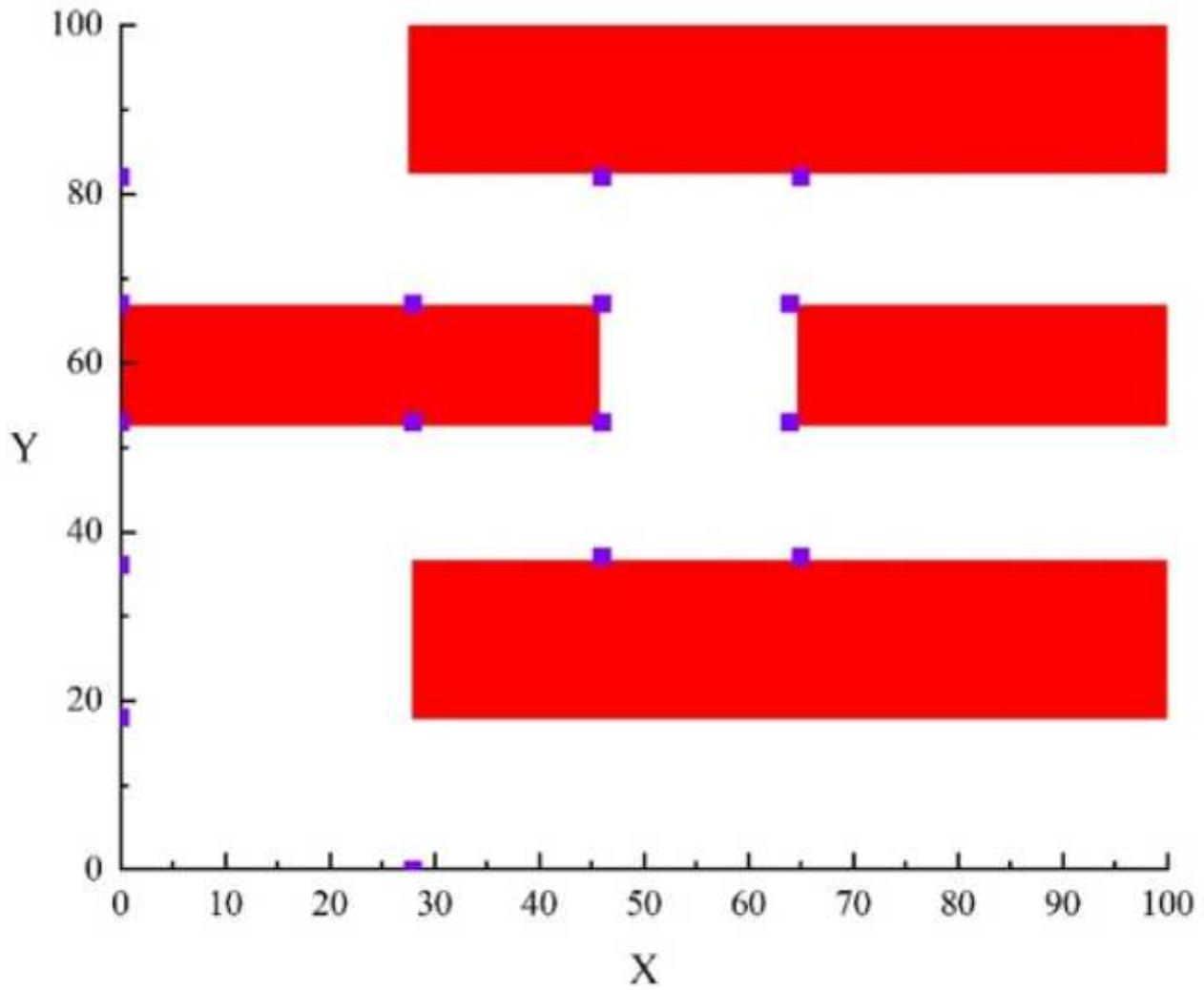


Figure 3

Results of the foot of perpendicular

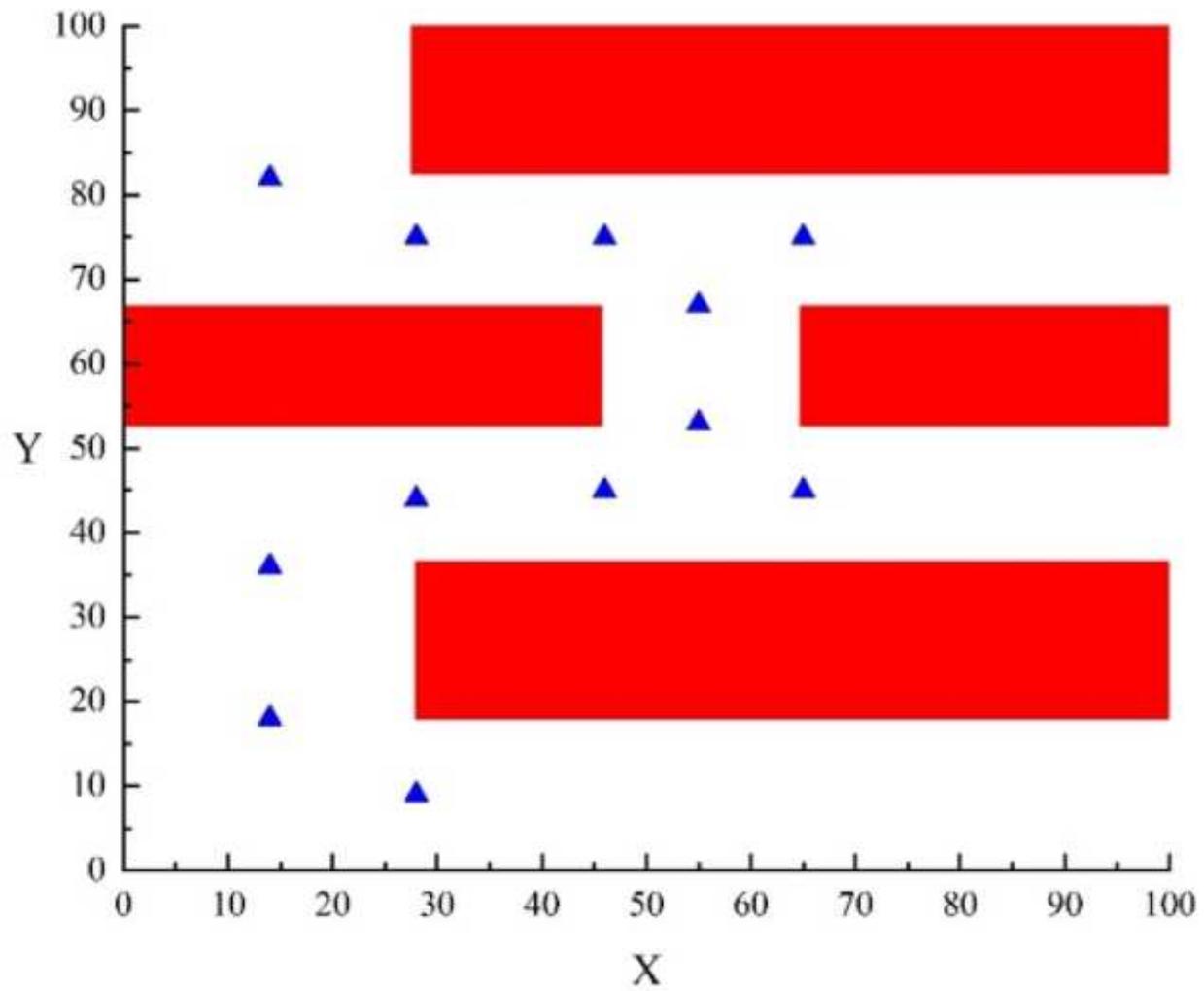


Figure 4

Midpoint detection results

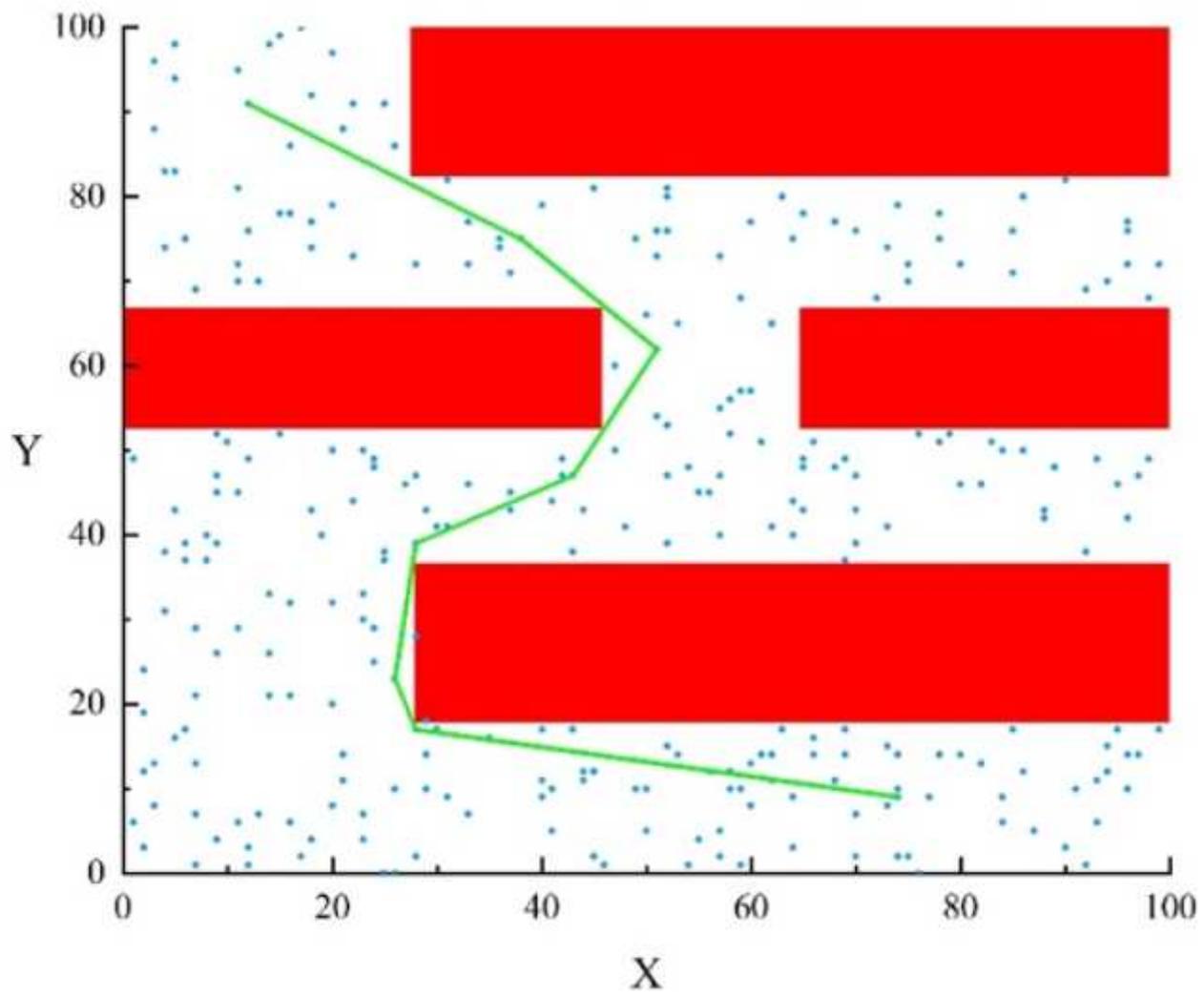


Figure 5

Ordinary PRM algorithm

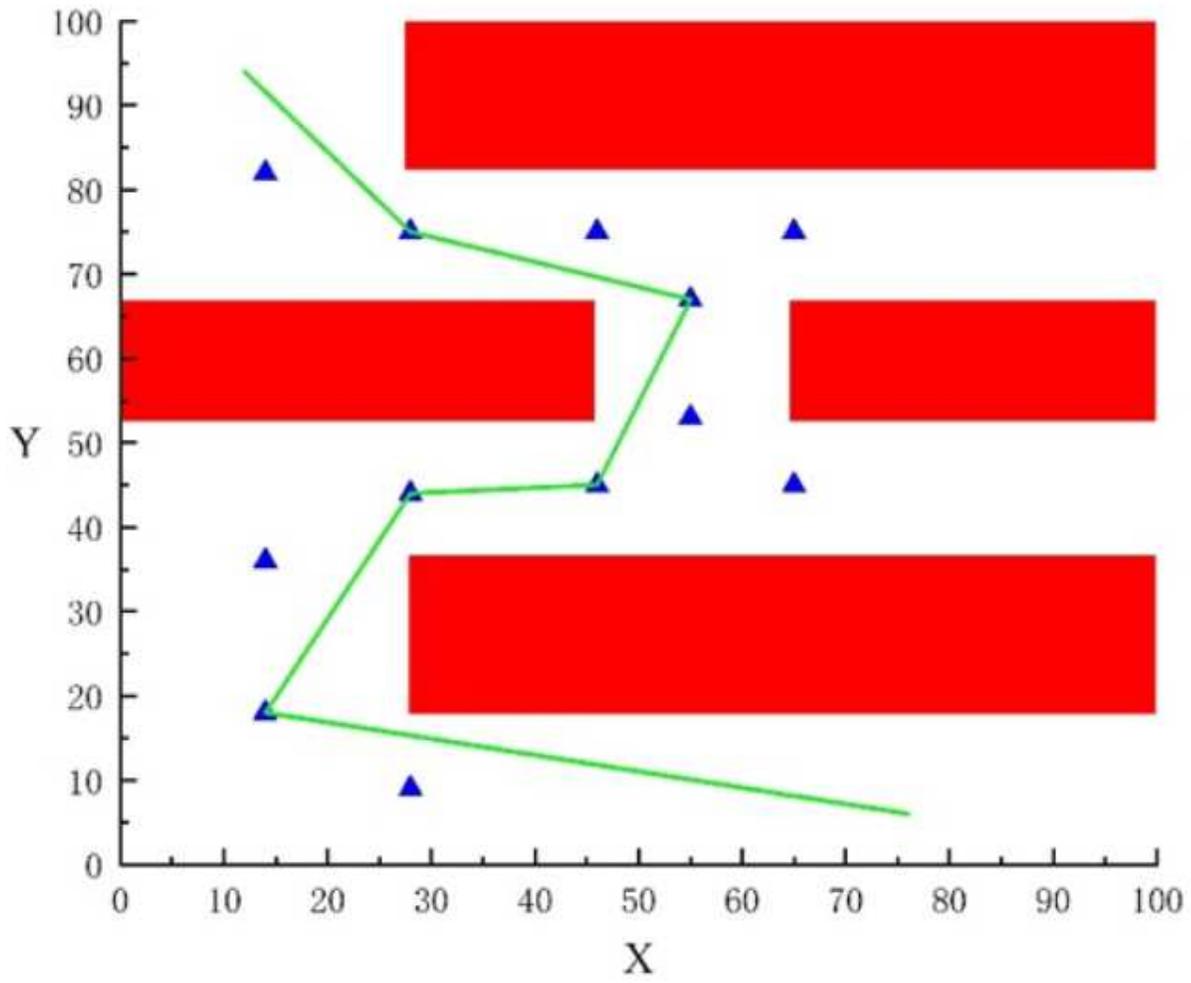


Figure 6

PRM algorithm using midpoint coordinates

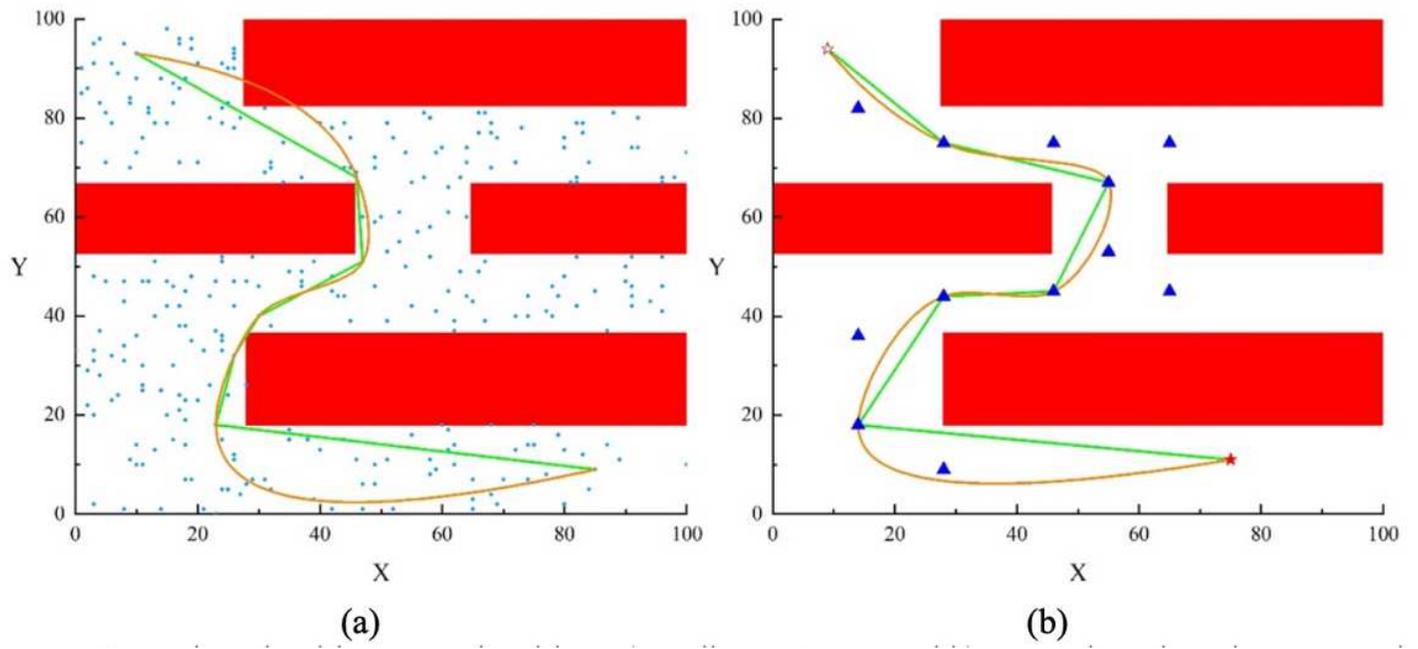


Figure 7

Smooth path with QPMI algorithm. a) Ordinary QPMI and b) QPMI based on the proposed method