2021 IEEE International Conference on Robotics and Automation (ICRA 2021)
May 31 - June 4, 2021, Xi'an, China

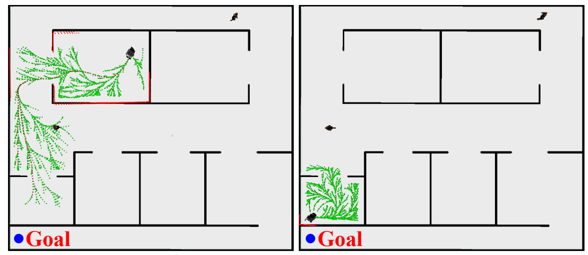
# A Knowledge-Based Fast Motion Planning Method Through Online Environmental Feature Learning

Yuan Yuan, Jie Liu, Jiankun Wang, Wenzheng Chi*, Guodong Chen* and Lining Sun*

***Abstract*— The sampling-based partial motion planning algorithm has come into widespread application in dynamic mobile robot navigation due to its low calculation costs and excellent performance in avoiding obstacles. However, when confronted with complicated scenarios, the motion planning algorithms are easily caught in traps. In order to solve this problem, this paper proposes a knowledge-based fast motion planning algorithm based on Risk-RRT, which guides motion planning by constructing a topological feature tree and generating a heuristic path from the tree. Firstly, an online topological feature learning method is proposed to simultaneously extract the features during the motion of the robot by means of the dual-channel scale filter and the secondary distance fusion. The learning process is completed until the feature points can represent arbitrary obstacle-free grid points of the whole map. Secondly, the topological feature tree is constructed with environmental feature points and the heuristic motion planning can be carried out on the feature tree. For one map, once the construction of the feature tree finishes, it can be reused as a prior knowledge in the following heuristic motion planning process, which will further improve the efficiency of searching feasible paths. The experimental results demonstrate that our proposed method can remarkably reduce the time taken to find a heuristic path and enhance the success rate of navigation in trapped environments.**

## I. INTRODUCTION

Nowadays, motion planning in dynamic human-robot coexisting environment has attracted increasing attention in the field of robotics. The classical motion planning algorithms can be roughly divided as follows. The grid-based planning algorithms can obtain the optimal trajectory, but at the same time, its graph search process is time-consuming, such as the A* [1] and the Dijkstra's algorithm [2]. The artificial potential field-based planning algorithms [3] do not require a high amount of computation, but it can not avoid the local minimum problem, so the method is incomplete and non-optimal. The sampling-based local motion planning algorithms such as the Rapidly-exploring Random Trees (RRT) [4], the Probabilistic Road Maps (PRM) [5], and other variants [6] [7] have been widely employed in mobile robot navigation due to their superior performance of low calculation costs, flexible obstacle avoidance and modeling-free of the environment [8] [9] [10]. The typical procedures of the sampling-based methods usually include node sampling, node selection, node connection and rewiring. In some complex human-robot coexisting environments, the robot often falls into the trap space. These trapped scenarios can be caused by the complicated static map, such as maze, S-bend and the slit, and it can also be generated by the flow of crowds. In such scenario, the node connection is rarely successful in these trapped regions, resulting in longer path planning time and even planning failure. As shown in Fig. 1, the robot is stuck in the local minimum in the trap space. The robot may occasionally plan successfully in the trap space, however, this knowledge is not retained by the robot. When receiving similar tasks, the robot still have to re-plan since it cannot learn from previous successful experiences, therefore, it is still easy to get stuck. In contrast, humans are good at summing up knowledge from previous successful experiences and utilizing the knowledge to find an appropriate way.




（a） t =1s （b） t =60s

Fig. 1. The trap space problem. Robot falls into the trap and can not escape.

Inspired by the motion planning scheme of humans, we propose a knowledge-based fast motion planning algorithm based on Risk-RRT, which guides motion planning by constructing a topological feature tree from its planning experience and generating a heuristic path [11] [12] from the information tree. An online feature learning algorithm is proposed to extract features that represent the environmental structure by using the information perceived during the motion of the robot. Inspired by Rapidly-exploring Random Trees (RRT) [13], we propose the topological feature tree to organize the feature points for rapidly searching. When the feature points are capable of representing arbitrary obstacle-free grid points of the whole map, the learning process finishes. Then the topological feature tree is constructed and the heuristic path is generated. The nodes on the heuristic

* corresponding author

This project is partially supported by NSFC grant #61903267 and National Key R&D Program of China grant #2019YFB1310003 awarded to Wenzheng Chi.

Yuan Yuan, Jie Liu, Wenzheng Chi, Guodong Chen and Lining Sun are with the Robotics and Microsystems Center, School of Mechanical and Electric Engineering, Soochow University, Suzhou 215021, China {jliu, yyuan, wzchi, gdchen, lnsun}@suda.edu.cn

Jiankun Wang is with the Department of Electronic and Electrical Engineering of the Southern University of Science and Technology in Shenzhen, China {wangjk}@sustech.edu.cn

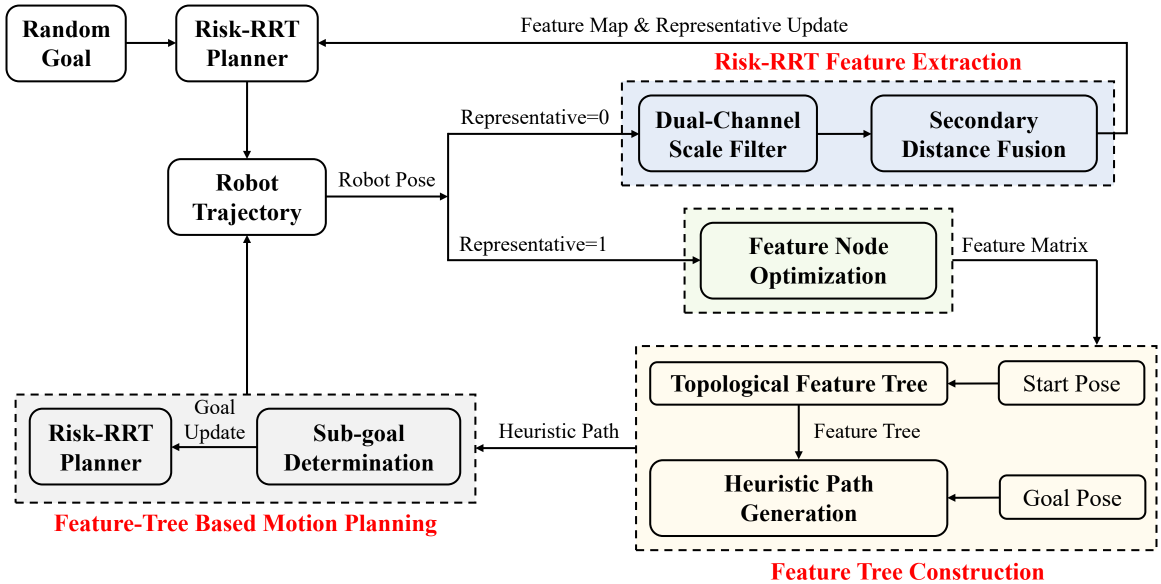



Fig. 2. The system architecture of the knowledge-based fast motion planning method.

path are then utilized as sub-goals to guide the sampling process for the RRTs.

The contributions of our work are summarized as follows:

- A novel knowledge-based fast robot motion planning framework addressing the trap space problem;
- An online feature extraction algorithm cooperating the dual-channel scale filter and the secondary distance fusion; and
- A topological feature tree construction algorithm for rapid feature searching.

The remainder of this paper is organized as follows: In Section II, we introduce the related work. In Section III, we explain the details of the proposed online feature learning algorithm. We conduct a series of experimental studies and discuss the results in Section IV and finally draw conclusions in Section V.

## II. RELATED WORK

Many algorithms have been put forward to solve the trap space problem in the field of the sampling-based motion planning. Liu *et al.* [14] propose an improved RRT algorithm by introducing a human-assisted decision making strategy where a virtual target point is set manually to guide the node searching out of the trap space. Moreover, a fast convergence strategy is also proposed to delete redundant nodes to speed up the planning process. Dong *et al.* [15] add the gravitational potential field as the prior environment information in the Q-learning algorithm and the Q value iteration of the concave trap area is eliminated, which increases the convergence speed of path planning. The algorithm can effectively avoid obstacles in the initial state and is suitable for direct learning in the real environment. Since the artificial potential field method is easy trapped in local minima, Xu *et al.* [16] propose the rotation velocity vector angle to accurately locate the escape point, and a TAS-RRT algorithm is combined with the artificial potential field algorithm for dynamic path planning. Gammell *et al.* [17] present the Batch Informed Trees (BIT*) algorithm, which combines the advantages of sampling-based planning algorithm with the grid-based path planning algorithm [1] to limit the sampling area of batches. Wang *et al.* propose the EB-RRT* [18] algorithm to guarantee the optimality of the path and prevent the trap problem on the basis of the RRT-Connect algorithm and the B-RRT* algorithm. These methods definitely improve the performance of the sampling-based motion planning algorithm in trapped environments, which can be applied successfully in the knowledge-free motion planning.

Another research direction to solve the trap problem is the knowledge-based motion planning. Herein, the knowledge means the prior information extracted from the previous experience, such as planning, perception, etc. Wang and Meng [19] propose to utilize Generalized Voronoi Diagram (GVD) of the environmental map to guide the sampling process. Chi *et al.* [20] propose a reusable generalized voronoi diagram feature tree algorithm based on [21] [22] to represent the topological structure of the map and generate a heuristic path to guide the motion planning. However, extracting GVD points is also time-consuming and this procedure is usually executed off-line.

In practical applications, mobile robots usually need to repeatedly execute motion planning on the map when performing different tasks. In this process, the robot can perceive considerable information, which can be utilized for further motion planning. Inspired by this idea, we present an online feature learning method to extract the topological information during the motion of the robot. A typical RRT variant for the dynamic environment, namely the Risk-RRT algorithm [23], is adopted to generate paths for the feature learning. We apply a tree structure to connect the feature points. When given the start and the end point, a heuristic path can be generated directly from the topological feature tree, which is utilized to guide the node sampling of the Risk-RRT algorithm in turn. In brief, the proposed method equips the robot with a brain memory and stores the prior knowledge in it. The more it is used, the more effective it will be.

## III. ALGORITHMS

The system architecture of the proposed knowledge-based fast motion planning method is illustrated in Fig. 2. To begin with, when given a brand-new environmental map, there is no feature nodes to represent the topological information and thus the representative value is set as 0. When the robot receives a goal, the robot will directly utilize the Risk-RRT planner to generate a feasible path. As the movement of the robot, the preliminary feature nodes are simultaneously extracted from the poses of the robot. As shown in Fig. 3, the trajectory of the robot is denoted with the blue dotted line, where the red flag represents the extracted feature points and the black rectangle represents the obstacle area. Then, a dual-channel scale filter and a secondary distance fusion strategy are utilized for further feature selection. On the basis of extracted feature nodes, we can update feature map $M_F$ and the feature matrix $M_P$. The feature map $M_F$ stores the index of the corresponding feature node for each grid point on the map. The feature matrix stores the connections of the feature nodes and the distance between two node neighbors.

When each obstacle-free grid point on the map has a corresponding feature node on the feature map, the representative value is set as 1 and the feature extraction process is completed. After that, when the start pose and the goal pose are given, the topological feature tree is constructed instantly, and the heuristic path is generated on the tree. Finally, we only need to sequentially take the nodes on the heuristic path as the sub-goals to guide the sampling of the Risk-RRT planner.

### A. Topological Feature Learning

In this part, we explain the topological feature learning method, which can be divided into three stages, namely the feature extraction and filtering, the feature topology construction and the feature optimization.

*1) **Dual-Channel Feature Filter**:* Firstly, the robot starts path planning with random goals based on the Risk-RRT algorithm, and the position and angular velocity of the robot can be obtained, which is recorded as $G_m = (x_m, y_m, \omega_m)$, where $x$ and $y$ represent the position of the node, and $\omega$ indicates the angular velocity of the node. In this part, we propose a dual-channel scale filter to select feature nodes from the original information $G_m$. The first channel is the angular velocity channel. In the angular velocity channel, the nodes with the angular velocity smaller than a manually set angular velocity threshold are filtered out. Herein, we only extract the nodes with a large angular velocity, since a large angular velocity indicates that the robot needs to turn rapidly at the current node. In other words, a small angular velocity indicates that the robot has been walking in a nearly straight line, and generally it can be directly connected to a previous feature node. The feature node screened out by the angular velocity channel is denoted as $G_{new} = (x_n, y_n)$. Moreover, another channel, namely the Euclidean distance channel, is also adopted with a distance threshold. When a new feature node is extracted from the angular velocity channel, we calculate the Euclidean distance between $G_{new}$ and all the nodes in the feature set $G_{PF}$. If the distance is larger than the pre-designed threshold, the node $G_{new}$ is added to the feature set, otherwise it is neglected.

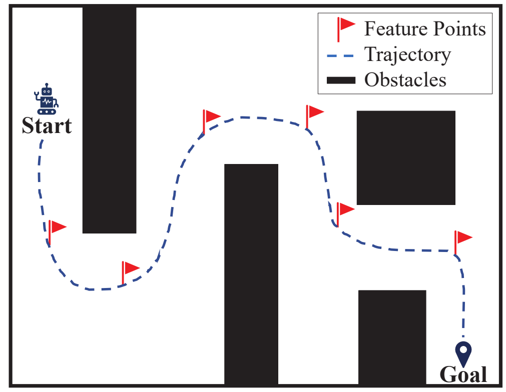



Fig. 3. Schematic diagram of feature node extraction.

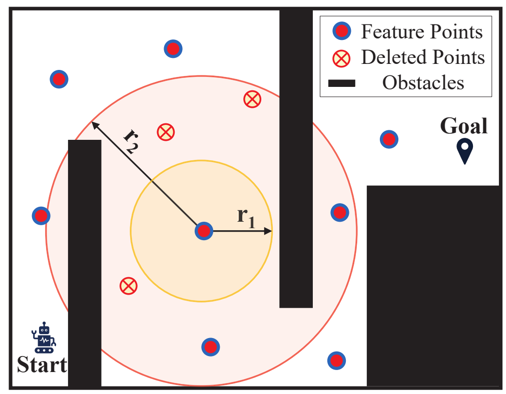



Fig. 4. The secondary distance fusion.

*2) **The Secondary Distance Fusion**:* In order to further reduce the redundancy of feature nodes, we propose a secondary distance fusion method to refine the feature nodes. Firstly, we calculate the number of neighbors of each feature point in $G_{PF}$. All the feature points whose number of neighbors are less than or equal to 3 are extracted in the reserved feature set $G_R$. The remaining nodes $G_D = G_{PF} \backslash G_R$ will participate in the secondary distance fusion. Secondly, we start from the first node, namely $(G_D)_k$. If the distance between $(G_D)_k$ and $(G_D)_{k+1}$ is less than the secondary fusion distance threshold and they pass the collision checking, we remove $(G_D)_{k+1}$ from $G_{PF}$ and return to the second step to begin a new iteration with $(G_D)_{k+1}$. As shown in Fig. 4, $r_1$ represents the threshold of the first distance fusion while $r_2$ indicates the threshold of the secondary distance fusion. The red cross represents the feature points fused by the secondary distance. When all the iterations end, we can obtain a concise feature set $G_{PF}$.

*3)* ***Feature Map, Feature Matrix and Feature Representativeness****:* After the feature extraction, a mapping process from the grid on the feasible area of the map to the extracted feature node is carried out, which finds the nearest feature node that the grid points can connect to without any collision. A feature map is proposed to record the correspondence between the grid point and the feature node. Each feature node usually responds to several grid points surrounding it. The feature map $M_F \in \mathbb{R}^{M \times N}$ is a two-dimensional matrix of the same size as the environmental map $M \in \mathbb{R}^{M \times N}$. Each value on the feature map records the index of its corresponding feature node. Therefore, we can directly retrieve the feature node of an arbitrary grid point $(i,j)$ by a simple mapping $(i,j) \rightarrow M_F(i,j)$.

A feature matrix [24] is utilized to represent topological connections among feature nodes and $F_G \in \mathbb{R}^{K \times K}$, where $K$ is the number of the extracted feature nodes. $F_G(u,v)$ denotes the topological connection between the *u-th* feature node and the *v-th* feature node. If they can be connected without collision, $F_G(u,v)$ records the inter-feature distance; otherwise $F_G(u,v)$ is set as zero. The advantage of the feature matrix structure is that it can quickly retrieve the neighbors of an arbitrary feature node, which is adopted to speed up the heuristic path planning process (seeing Section III-B in detail).

Herein, the representativeness of feature nodes is calculated, namely the ratio of grid points which obtain feature nodes to total grid points on the map. Initially, the representative value is set as 0 and it increases during the feature learning process. When the representative value reaches 1, it means that all the grid points on the feasible area can be connected to at least one feature node. In other words, the feature nodes can represent the topological information well.

*4)* ***Feature Node Optimization****:* As mentioned above, when the representative value equals to one, it means the number of feature points is enough to complete the representation of the map, and it is no longer necessary to add feature points. At this stage, a feature node optimization strategy is proposed to refine the original feature points. Firstly, the Euclidean distance between the new generated feature node $G_{new}$ and a node $G_i$ in the current feature set is calculated and the feature node with the minimum distance is found by:

$$I_{min} = \arg\min_{i=1}^{K} ||G_{new} - G_i||, \tag{1}$$

where $K$ denotes the total number of feature nodes in the current feature set. Then the grid points on the map that correspond to the feature node $G_{I_{min}}$ can be retrieved from the feature map. Herein, we use $\mathcal{G}_{I_{min}}$ to denote these grid points. Next, the collision checking is conducted between the grid points in $\mathcal{G}_{I_{min}}$ and $G_{new}$. If all grid points pass the collision checking, then we move to the next stage, otherwise the optimization process ends.

In the second stage, the average distance from the feature node to its corresponding grid points is utilized to evaluate the feature node, which is defined as:

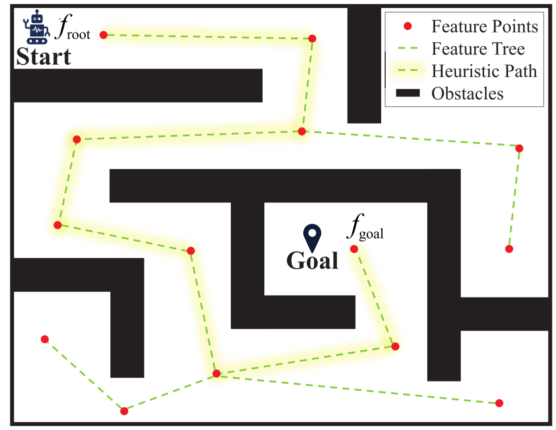



Fig. 5. An illustration of the heuristic path planning.

$$d(f_k) = \frac{1}{count(\mathcal{G}_{I_{min}})} \sum_{g_i \in \mathcal{G}_{I_{min}}} ||f_k - g_i||, \tag{2}$$

where $count(\mathcal{G}_{I_{min}})$ denotes the total number of the grid points in $\mathcal{G}_{I_{min}}$. If $d(G_{new}) < d(G_{I_{min}})$, $G_{I_{min}}$ will be replaced with $G_{new}$. This process will be iterated all the time, and the more plans the robot has made on the same map, the more representative the feature points will be.

### *B. Fast Motion Planning with the Topological Feature Tree*

In this part, we explain the motion planning process with the learned feature set (representative = 1), which can be divided into two stages, namely the heuristic path planning and the partial motion planning. After the feature learning process, the robot has obtained a prior knowledge of the environment, namely the feature node set $G_F$, the feature map $M_F$, and the topological connections among features $F_G$. In the first stage, a heuristic path is generated by using the prior knowledge. An illustration of the heuristic path planning is shown in Fig. 5. The specific procedure is presented as below.

- ***Feature Mapping***. When the start pose is given, $(i_s, j_s)$ for example, its corresponding feature node can be retrieved immediately from the feature map $(i_s, j_s) \rightarrow M_F(i_s, j_s)$. This feature node is chosen as the root $f_{root}$ of the feature tree.
- ***Neighbor Retrieving***. Suppose the index of $f_{root}$ be $i_{root}$. The neighbor feature nodes of $f_{root}$ can be then retrieved from the $i_{root}$-*th* row of the feature matrix $F_G$. We denote the neighbor nodes as $f_{neighbor}$.
- ***Node Connection***. The root node is set as the parent node and the neighbor nodes are connected to the parent node as children nodes. During the connection, the cumulative cost from the root to the current node is also calculated by:

$$c(f_k) = c(f_k \rightarrow parent) + F_G(f_k). \tag{3}$$

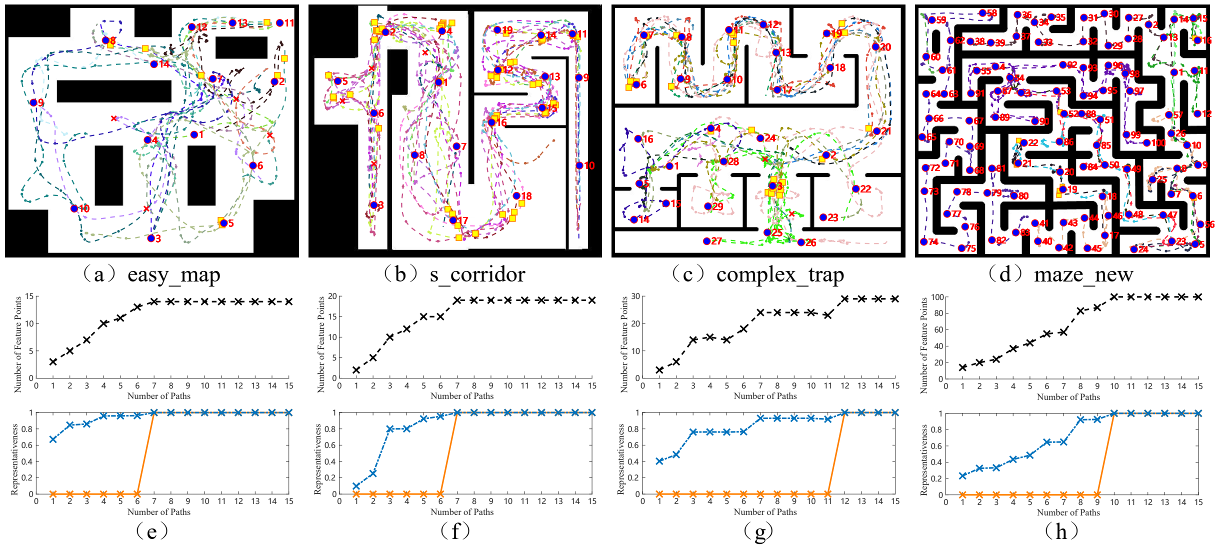



Fig. 6. Feature Learning Experiment. The dashed lines represent the trajectories the robot has perceived, the red cross represents the redundant feature nodes deleted by the secondary distance fusion, and the yellow square represents the replaced feature nodes according to the optimal average distance. The blue solid circles represent the final extracted feature nodes.

It is noteworthy that $F_G(f_k)$ returns the distance between the current node (parent node) and its neighbor node ($f_k$) only when there is no collision between them in the map. Therefore, by using the feature matrix, no collision checking among feature nodes is required in the connection process. In the next iteration, the neighbor nodes are set as the parent nodes in turn and their neighbors are set as the child nodes and connected to the tree.

- ***Node Rewiring***. When a node is already on the tree and a new connection route appears, we then compare the cost of the current connection and the new route. If the cost of the new route is less than the original one, the node is rewired with the new route.

When a goal pose is given, its corresponding feature node $f_{goal}$ is retrieved from the feature map. A heuristic path can be generated directly by a backward search from $f_{goal}$ to $f_{root}$, as denoted with yellow in Fig. 5. The feature nodes on the heuristic path are then utilized as sub-goals gradually for the Risk-RRT planner so as to guide the sampling in the trapped environment. We can find that the feature tree growth and the heuristic path generation process only involve simple calculations and have no time-consuming module such as the collision checking. Therefore, it can be executed in real time.

## IV. EXPERIMENTAL STUDIES AND RESULTS

In this work, we carry out experimental studies on the basis of Robot Operating System (ROS) and a computer with Ubuntu 16.04 on an Intel i7-9750H CPU with 8GB of RAM is adopted as the simulation platform. Firstly, the performance of the feature learning process is tested on map with different complexity. Followed by the efficiency and effectiveness examination of the proposed fast motion planning algorithm with the original Risk-RRT as a reference in the human-robot coexisting environment, especially in the complex trap space. In the motion planning experiments, we record the navigation success rate, the navigation running time, and the length of the entire paths for evaluations.

### A. The Performance in Feature Learning

During the feature learning, we record the number of extracted feature points and the corresponding representative value of the feature set. Four environmental maps with different complexity have been selected, namely *easy_map*, *s_corridor*, *complex_trap* and *maze_new*, as illustrated in Fig. 6(a)-Fig. 6(d). The dashed lines represent the trajectories the robot has traveled; the red cross represents the redundant feature nodes deleted by the secondary distance fusion; and the yellow square represents the replaced feature nodes during the feature optimization. The solid blue circles represent the final extracted feature nodes. As demonstrated in Fig. 6(e)-Fig. 6(h), at the beginning, as the number of paths traversed by the robot increases, the number of topological feature points (black line) increases and the representative value (blue line) increases. When the representative value rises to 1, the number of feature points no longer increases. In the motion planning process, the representative value less than 1 is set as 0 for calculation convenience, as denoted with orange lines.

### B. The Performance in Motion Planning

The experiments on robot navigation are carried out with the aforementioned four maps. Fig. 7(a)- Fig. 7(d) show the trajectories generated by the Risk-RRT planner while Fig. 7(e)- Fig. 7(h) show the trajectories generated by our method. In these scenarios, the continuous blue circles represent pedestrians.

In the *easy_map* scenario, four pedestrians are included. This is the simplest map where there is no trap space. We can find that the performance using the Risk-RRT is similar

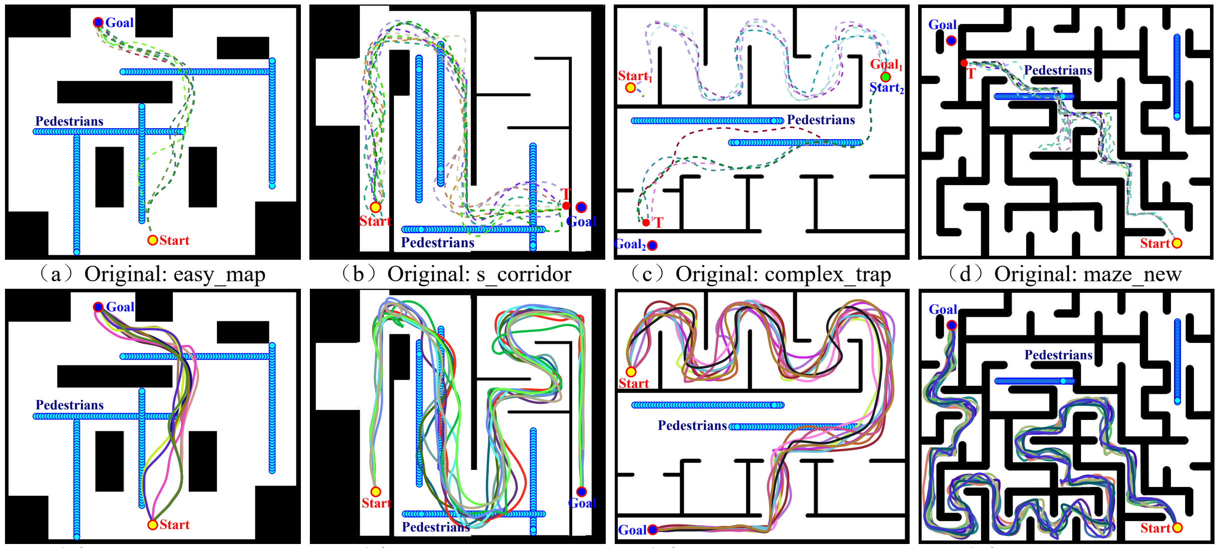



Fig. 7. Trajectories generated with/without the heuristic paths generated by the proposed method. The upper shows the performance with the original Risk-RRT planner and the lower shows the performance of the proposed method.

TABLE I
THE STATISTICS OF THE MOTION PLANNING EXPERIMENTS

| Scenario | | | Risk-RRT | | | Ours+Risk-RRT | | |
|---|---|---|---|---|---|---|---|---|
| Environment | Map Size | Pedestrians | Success Rate | Time(s) | Length(m) | Success Rate | Time(s) | Length(m) |
| easy_map | 50.4m*43.4m | 4 | 10/10 | 33.31±2.26 | 26.95±1.21 | 10/10 | 32.34±2.46 | 26.36±0.58 |
| s_corridor | 50.4m*43.4m | 4 | 0/10 | - | - | 10/10 | 138.26±23.20 | 95.90±4.64 |
| complex_trap | 50.4m*43.4m | 2 | 0/10 | - | - | 10/10 | 122.68±16.39 | 84.20±3.01 |
| maze_new | 50.4m*43.4m | 2 | 0/10 | - | - | 10/10 | 191.02±3.29 | 287.63±12.71 |

to that of our proposed algorithm, as shown in Fig. 7(a) and Fig. 7(e). In the $s\_corridor$ scenario that also includes four pedestrians, we specially set the s-bend and narrow corridor to create a trap space. We can find the path generated by the Risk-RRT is prone to be trapped in the local minima $T$, as shown in Fig. 7(b). However, with our proposed method, the robot can easily navigate to the target point, as shown in Fig. 7(f). In the $complex\_trap$ scenario including two pedestrians, the robot is trapped in a trap from the very beginning with the Risk-RRT planner, so we add another $Goal_1$ to guide its planning and then set the final $Goal_2$. The robot still fell into the trap for the difficulty on sampling in the narrow corridor, as shown in Fig. 7(c). In contrast, our method shows a superior performance, as shown in Fig. 7(g). In the complicated $maze\_new$ scenario containing two moving pedestrians, the path guided by the Risk-RRT planner tends to be trapped in the local minima $T$, as shown in Fig. 7(d) while our method can successfully reach the goal, as shown in Fig. 7(h).

Table I lists the statistics that we obtain from ten times of repeated trials. The time here denotes the runtime of the navigation from the start pose to the goal pose. We can find that with the proposed method, the success rate of the navigation increases obviously. In conclusion, the experimental results demonstrate that our proposed algorithm can solve the trap space problem successfully in complicated human-robot coexisting environment.

## V. CONCLUSIONS

In this paper, we have proposed a knowledge-based fast motion planning algorithm by constructing a topological feature tree and generating a heuristic path from the tree. A dual-channel feature filter and a secondary distance fusion method have been presented to extract topological features from the previous paths of the robot. The feature nodes, feature map and feature matrix formulate the prior information for the robot, which achieves a concise topological representation of the environmental map. A heuristic path planning is proposed by constructing a feature tree with the prior information. A fast motion planning algorithm is proposed by integrating a heuristic path with the sampling-based planner. Experiments on both feature learning and motion planning reveal that our method performs well on maps with different complexity and solves the trap space problem effectively.

## REFERENCES

[1] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.
[2] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, no. 1, pp. 269–271, 1959.
[3] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *Autonomous Robot Vehicles*. Springer, 1986, pp. 396–404.
[4] S. M. LaValle, "Rapidly-exploring random trees: A new tool for path planning," 1998.

[5] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars, “Probabilistic roadmaps for path planning in high-dimensional configuration spaces,” *IEEE Transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.

[6] C. Wang, W. Chi, Y. Sun, and M. Q. . Meng, “Autonomous robotic exploration by incremental road map construction,” *IEEE Transactions on Automation Science and Engineering*, vol. 16, no. 4, pp. 1720–1731, 2019.

[7] W. Chi, C. Wang, J. Wang, and M. Q. . Meng, “Risk-dtrrt-based optimal motion planning algorithm for mobile robots,” *IEEE Transactions on Automation Science and Engineering*, vol. 16, no. 3, pp. 1271–1288, 2019.

[8] S. Petti and T. Fraichard, “Safe motion planning in dynamic environments,” in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2005, pp. 2210–2215.

[9] L. Jaillet, J. Cortés, and T. Siméon, “Sampling-based path planning on configuration-space costmaps,” *IEEE Transactions on Robotics*, vol. 26, no. 4, pp. 635–646, 2010.

[10] D. Devaurs, T. Siméon, and J. Cortés, “Optimal path planning in complex cost spaces with sampling-based algorithms,” *IEEE Transactions on Automation Science and Engineering*, vol. 13, no. 2, pp. 415–424, 2016.

[11] D. Shaharabani, O. Salzman, P. K. Agarwal, and D. Halperin, “Sparsification of motion-planning roadmaps by edge contraction,” in *2013 IEEE International Conference on Robotics and Automation*, 2013, pp. 4098–4105.

[12] Dobson, Andrew, Bekris, and E. Kostas, “Sparse roadmap spanners for asymptotically near-optimal motion planning.” *International Journal of Robotics Research*, 2014.

[13] S. Lavalle and J. Kuffner, “Rapidly-exploring random trees: Progress and prospects,” *Algorithmic and computational robotics: New directions*, 01 2000.

[14] L. Huawei, Z. Shuai, Z. Boxin, and Z. Xiaolin, “Uav path planning based on improved rrt algorithm in trap space,” *Computer Engineering and Applications*, 2018.

[15] D. Peifang, Z. Zhian, M. Xinhu, and Z. Shuo, “Reinforcement learning path planning algorithm based on gravitational potential field and trap search,” *Computer Engineering and Applications*, 2018.

[16] X. Xiaohui and Z. Jinlong, “Hybrid optimization algorithm of improved artificial potential field and tas-rrt,” *Application of Electronic Technique*, 2018.

[17] J. D. Gammell, S. S. Srinivasa, and T. D. Barfoot, “Batch informed trees (bit*): Sampling-based optimal planning via the heuristically guided search of implicit random geometric graphs,” in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 3067–3074.

[18] W. Kun, Z. Guohui, L. U. Dunke, H. Bo, and L. I. Xiaobin, “Path planning of mobile robot based on improved asymptotically-optimal bidirectional rapidly-exploring random tree algorithm,” *Journal of Computer Applications*, 2019.

[19] J. Wang and M. Q. . H. Meng, “Optimal path planning using generalized voronoi graph and multiple potential functions,” *IEEE Transactions on Industrial Electronics*, vol. 67, no. 12, pp. 10 621–10 630, 2020.

[20] W. Chi, J. Wang, Z. Ding, G. Chen, and L. Sun, “A reusable generalized voronoi diagram based feature tree for fast robot motion planning in trapped environments,” *IEEE Sensors Journal*, pp. 1–1, 2021.

[21] M.-L. Shao, J.-Y. Lee, C.-S. Han, and K.-S. Shin, “Sensor-based path planning: The two-identical-link hierarchical generalized voronoi graph,” *International Journal of Precision Engineering and Manufacturing*, vol. 16, pp. 1883–1887, 07 2015.

[22] Q. Wang, M. Langerwisch, and B. Wagner, “Wide range global path planning for a large number of networked mobile robots based on generalized voronoi diagrams,” *IFAC Proceedings Volumes*, 2013.

[23] C. Fulgenzi, A. Spalanzani, C. Laugier, and C. Tay, “Risk based motion planning and navigation in uncertain dynamic environment,” *INRIA, Research Report*, 2010.

[24] W. Chi, Z. Ding, J. Wang, G. Chen, and L. Sun, “GVD feature matrix based efficient heuristic path planning for mobile robots,” *Transactions on Industrial Electronics (Under Review)*, 2020.