









# Evaluation of Motion Planning Algorithms for Underground Mobile Robots

Desheng Zhu<sup>1</sup> , Yuanyuan Zhang<sup>1</sup> , Jiawen Wang<sup>1</sup> , Kaiqi Ren<sup>2</sup> ,  
and Kehu Yang<sup>1</sup>  

<sup>1</sup> School of Mechanical Electronic and Information Engineering, China University of Mining and Technology, Beijing, China  
ykh@cumtb.edu.cn

<sup>2</sup> Industrial and Commercial Bank of China, Beijing, China

**Abstract.** The underground coal mine is an extremely hard environment for workers due to geological hazard, poisonous gas, and harmful dusts. In such environment, coal mining robots are very helpful for coal mines to reduce the number of onsite workers, enhance safety, and improve efficiency. However, the autonomous movement of coal mine robots in the underground tunnel is still a big challenge. This paper gives a detailed evaluation of the commonly used motion planning algorithms for underground mobile robots. Firstly, the motion planning framework for mine mobile robots is proposed. It considers the global path planning, local path planning, and the recovery behavior. Then, a motion testing framework in a simulated coal mine environment is constructed based on the ROS platform to test the commonly used motion planning algorithms, such as the the Dijkstra, A\*, rrt\* and Hybrid A\* algorithms. The test results shows that RRT\* of DWA algorithm runs faster than other algorithm but Hybrid A\* of DWA algorithm gets the shortest and smoothest path.

**Keywords:** Motion planning · Coal mine robot · Evaluation · ROS

## 1 Introduction

Coal is the most important energy in China, which accounts for 56% of the total primary energy consumption in 2021. Coal consumption is expected to grow at a compound annual growth rate of 2.3 % between 2021 and 2025 to reach 5.6 billion in 2025 [1]. However, the labor shortages due to aging population become the biggest challenge in the mining industry. To solve the problem, intelligent robots can be used in a variety of mining scenarios, such as tunnelling, excavating and inspection [2]. Consequently, the use of robotic devices powered by artificial

---

This work is supported by National Natural Science Foundation (NSFC) of China under Grant 61973307 and 61936008.

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2022  
L. Zhang et al. (Eds.): CINT 2022, CCIS 1714, pp. 368–379, 2022.  
[https://doi.org/10.1007/978-981-19-8915-5\\_32](https://doi.org/10.1007/978-981-19-8915-5_32)

intelligence will transformed the coal industry, where safety and efficiency are the main issues.

The basic function for a coal mining robot is autonomously and agilely in the underground tunnel [3]. The autonomous movement of robots is a comprehensive engineering problem including perception, localization, planning, and control, shown as Fig. 1. The coal mine robot gets the environmental information of the mining space through cameras or lidars, and position and orientation of robot are calculated by positioning fusion algorithms such as Kalman filtering. By calculating a passable path used the motion planning algorithm, the robot is able to avoids obstacles such as hydraulic struts on the coal mine face, and sends motion commands to the motion controller for execution [4].

Coal mine is make up of curvy, irregular tunnel, meanwhile obstacles are various and complicate. Due to the special underground tunnel environment, mining robots need to travel in complex and rugged areas, making motion planning become a difficult problem. Many mature algorithms commonly used by ground robots may be not suitable for coal mine environment, and never be experimental tested before. This paper mainly focuses on the research on motion planning of the coal mine robot.

The paper is organized as follows. Related work was given in Sect. 2. The motion planning framework illustrated in Sect. 3. Then, in Sect. 4, it was shown how to test the real robot in simulated coal mine environment and evaluated the operation effect of various motion planning algorithms. Finally, the experiment results was analyzed in Sect. 5.

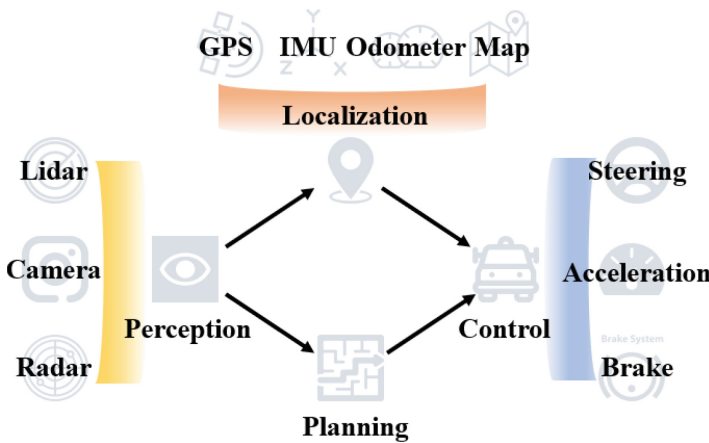


Fig. 1. The framework of autonomous movement

## 2 Related Work

Over the past few decades, many path planning technologies have been proposed, such as search-based method, sampling-based methods, potential field method and intelligent algorithm method.

Search-based method is a classic global path planning algorithm in the two-dimensional grid map. Representative algorithm include Dijkstra algorithm [5], A\* algorithm [6], Hybrid A\* [7] algorithm, D\* algorithm [8], etc. Among them, A\* is an improved version of Dijkstra, aimed at solving Dijkstra's inefficiency. Hybrid A\* algorithm is combination of A\* algorithm and vehicle kinematics to deal with kinematic constraint. D\* is Dynamic A Star to apply at dynamic environment. a two level A-path planning calculation method to overcome computation complexity has been proposed [9].

Sampling-based methods is path planning algorithm based on graph structure. Representative algorithm include Probabilistic Roadmap Method (PRM), Rapidly-Exploring Random Trees (RRT), and their improved algorithm RRT\* [10]. These algorithms find not relatively optimized paths but feasible paths.

The artificial potential field method is an collision avoidance approach, which the obstacle gives the robot a repulsive force and the target points give it attractive force. However, the artificial potential field method suffers from the local minimum and goal non-reachable with obstacles nearby problem. A improved artificial potential field method has been proposed to make mining robot go out of local minimum point autonomously [11]. Y. Lei proposed a fuzzy logic-based adaptive DWA, which considering obstacle avoidance is designing by taking Ackermann steering constraint into account [12].

But those algorithms commonly may be not suitable for coal mine environment, and need to be tested and evaluated practically.

## 3 Motion Planning Framework

The motion planning problem can be described as looking for the optimal motion trajectory of the robot from the initial state to the target state. Generally, it also needs to meet some motion constraints [13]. For example, they need to avoid the fixed mining machines and the moving miners in the narrow tunnel. The constrained planning problems also maintain constraints corresponding to motion with a bounded turning angle. This section will discuss global planners, local planners, and recovery behaviors in sequence.

Many mature algorithms have been applied to a variety of robots [14]. Our algorithms adopt the framework of global planner, local planner, and recovery behaviors, shown as Fig. 2:

- **Global planner:** Through global planner, the robot has gained a map to plan out a feasible path roughly, known as the global route.
- **Local planner:** Within the scope of the local sensor detection, local planner detects dynamic obstacles and actual traffic area around the robot's motion, as well as follows control rules and global path to calculate a local path.

- **Recovery behaviors:** Vehicle controller algorithm transforms the planning path into the control instruction. If the robot encounters an impassable path or gets lost, the recovery system is activated to reset the robot situation.

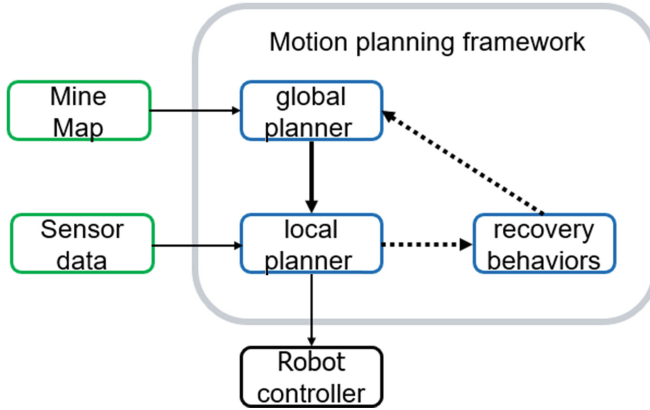


Fig. 2. The framework of motion planning

### 3.1 Global Planner

The global planner gives information about obstacles and environments contained in the map, the robot's position, and targets in the world. It creates a global path to reach the target position. The generated paths do not consider the dynamics and kinematics of the vehicle, resulting in an impassable path for the robot. Commonly used algorithms are Dijkstra, A\*, rrt\*, Hybrid A\*.

- **Dijkstra:** Dijkstra is a shortest path algorithm for searching one vertex to other vertices, which solves the shortest path problem in directed graph. The main characteristic of Dijkstra algorithm is that it starts from the starting point to traverse to the adjacent nodes of the nearest and unvisited vertex at the beginning point each time until it extends to the end point. The Dijkstra algorithm is breadth-first search, which is a kind of divergent search, so the space complexity and time complexity of Dijkstra algorithm are relatively high.
- **A\*:** A\* algorithm is classical heuristic search algorithm, which is the improvement algorithm from classical search algorithm Dijkstra. The most significant characteristic of the algorithm is that the heuristic function is given in the process of search to reduce the search nodes, thus improving the efficiency of path search. Beginning from the start point, A\* algorithm targets at the terminal point with iterating over the adjacent points around the start point

and the traversed starting point, and calculates the total movement cost from the starting point to each node.

In each iteration of the main loop, A\* needs to decide which paths to extend. It based on an estimate of the cost of the path and extending the path to the goal. Specifically, A\* selects the minimal path. A\* stop the operation when a path to expand from the start to the end, or when there are no paths to expand. If the heuristic function is properly, it absolutely does not overstate the actual cost of reaching the goal, then A\* ensures the least costly path back to the goal from the beginning.

- **RRT\***: RRT\* algorithm is a sampling-based path planning algorithm. An RRT\* generates sampling points randomly near the start point and connects the start point with the sampling point to generate a tree. After drawing the random tree, check that the path of the branch is passable. If the connection is feasible (completely through free space and with constraints), this results in the addition of the new state to the tree. Repeat the process until you have a path that leads from the beginning to the end.
- **Hybrid A\***: The Hybrid A\* algorithm considers the limitation of vehicle kinematic model in node expansion, and extends the 2-dimensional search to 3-dimensional space  $[x, y, \theta]$ , where  $\theta$  is the vehicle orientation, which can plan the continuous pose change of unmanned parking spaces in discrete grid. Based on Hybrid A\* algorithm, node expansion mode, collision detection, design of cost function and other aspects are improved to make the improved algorithm can faster search for the initial path.

### 3.2 Local Planner

The global path planner generates the general path of the vehicle motion without fast obstacle avoidance. The effect is not ideal when the vehicle directly executes the global path. On the one hand, the robustness of the global path planner is poor, reflected in the weak ability to avoid dynamic obstacles, and leads to planning on inaccurate maps. On the other hand, the search time of the global planner increases greatly with the search space. When searching in a high-dimensional search space, considering the increase in the amount of computation, the increase in computation time brought by re-planning is unbearable [15].

Local planners are responsible for getting trajectory to move the robot target position safely. The local planner tries to follow the global planner's plan while taking into account the kinematics and dynamics of the robot. To generate safe speed instructions, the local planner uses the Dynamic Window Approach (DWA) or Trajectory Rollout to simulate and select possible path according to the cost function.

- **Trajectory Rollout**: Trajectory Rollout discretely sample in the robot's control space, and generate a series of tracks. The set of trajectories is generated by changing the turning angle while maintaining a certain speed. The positions of trajectories are determined by calculating the turning angle every time step. After that, the Trajectory Rollout will identify which path

collision-free. In this case, the optimization algorithm elects the shortest path between the ideal future position and the final position of the trajectory.

- **DWA:** The principle of the DWA algorithm is to search in the state space, constrain the search space to a suitable range, generate multiple prediction paths, select the optimal path according to the path evaluation function, and send the state control command of the optimal path directly to the chassis implement.

Just like Trajectory Rollout, DWA searches for commands controlling the robot in the space of velocities. Firstly, the search space is reduced in circular trajectories, admissible velocities and dynamic window. Secondly, the trajectory is chosen which maximize objective function from the remaining trajectory. The objective function as:

$$G(v, \omega) = \alpha \cdot \text{heading}(v, \omega) + \beta \cdot \text{dist}(v, \omega) + \gamma \cdot \text{vel}(v, \omega) \quad (1)$$

*heading* represents the deviation towards target, and is the largest when the robot moves towards the target. *dist* is the distance to the nearest obstacle on the trajectory. The shorter the distance to the obstacle, the greater the desire of the robot to move around the obstacle. *vel* is the forward speed of the robot and supports high speed movement. The larger the value, the faster the movement speed.  $\alpha, \beta, \gamma$  denotes adjustable coefficient for heading, dist and vel, setting as defaults.  $v$  and  $\omega$  are the vehicle speed and orientation.

### 3.3 Recovery Behavior

The motion planning system can works well in most cases. However, when there are some dramatically changes in the mine, the robot need to reset. To make the system robust, the recovery system was built into the mine motion planning framework.

- **Rotation:** When the robot gets trapped, the robot rotates to scan for free space. If this fail, a more aggressive recovery behavior will be attempted.
- **Back to the start point:** The robot returns along the path that it has traveled, meanwhile, it robot reroutes based on real-time environment. When if find another path, it will execute it. If this fail, a more aggressive recovery behavior will be attempted.
- **Rebuild the map:** The robot will abandon the previous map and re-plan the path with the real-time surrounding environment map.

## 4 Performance of Simulated Environment Experiment

In this section, we present the setup, evaluation environment and detailed experiment results.

#### 4.1 Platform

Our experimental platform is built upon an Autolabor differential vehicle with an embedded computer(NVIDIA Jetson AGX Xavier), 2d Lidar(Slamtec RPLIDAR A2), high precision MEMS Inertial Sensor(ADIS16495), and 360 pulse per revolution wheel odometer attached in wheels.

The platform is driven by some subsystems: hardware drivers, controllers, perception, planning, higher-level control. We use ros to communicate between subsystems, which is a common communication framework for robots. Experiments in this paper were performed with an Autolabor Pro 1 mobile platform, The maximum speed of the Pro1 base is 0.8 m/s (Figs. 3 and 4).



**Fig. 3.** The overview of autonomous robot    **Fig. 4.** The static obstacle avoidance test

#### 4.2 Test Method of Coal Mine Simulation Conditions

Limited by the requirements of coal mine safety regulations, it is inconvenient to test on the actual site. However, to meet the actual production needs of coal mines, the production conditions of coal mines are imitated and 3 test experiments are set up to simulate the actual scene of coal mine roadways.

- **Roadway following test:** The operation of coal mine robot needs to reach the position accurately, so we carry out the inspection test of the task point. The task is to set the different target point in the corridor. The whole journey is 50 m in total. Every time the car reaches a target point, it will automatically

go to the path planned for the next target point. When it reaches a target point, it will be recorded as a successful navigation plan. The path released by the car and relevant data of operation are recorded.

- **Static obstacle avoidance test:** In the coal mine production environment, the single hydraulic prop is widely used in underground mining support equipment. The diameter of the commonly used single hydraulic prop cylinder is 10cm. As simulating a single hydraulic prop cylinder, paper cylinders of the same size are set in the middle to simulate the mining environment obstacles. The operation ability of various algorithms is tested in the narrow passages between the obstacles of the props.
- **Dynamic obstacle avoidance test:** In the coal mine environment, workers often move around when they at work, and the moving person is a dynamic obstacle that the robot must avoid. This paper takes the moving person as the moving obstacle and tests the obstacle avoidance capability.

### 4.3 Evaluation Metrics

For performance evaluation, we use the following metrics, shown as Table 1.

**Table 1.** Evaluation metrics

Evaluation metrics	Description
Success rate	The proportion of no intervention or collision in trial
Path length	Average length of the walking route when the task is successfully completed
Runtime	The whole time from sending the target point command to running to the final point
Global path planning time	The time between the acceptance of the target point and the release of the first planned route by the global planner
Number of Obstacle collision	Number of collisions during running tests including dynamic and static obstacles
Minimum/maximum allowable width	The minimum passing distance without collision/ no collision will occur through space that exceed this width
Temporal coefficient	The ratio of the actual time to the minimum travel time representing the time of planner execution
Spatial coefficient	Spatial coefficient is the ratio of the actual distance traveled to the planned distance representing the efficiency of local planner execution
Smooth coefficient	The ratio of the actual Angle turned to the accumulated Angle of the planned path representing the curvature of the path executed by the local planner

- **Success rate:** Success rate refers to the proportion of no intervention or collision in trial.
- **Path length:** The length of the path is the key to evaluating the quality of the route. is recorded as the average length of the walking route when the task is successfully completed, and if it cannot be completed, the longest distance traveled is recorded.
- **Runtime:** The running time is the whole time from sending the target point command to running to the final point.
- **Global path planning time:** The time between the acceptance of the target point and the release of the first planned route by the global planner.



- **Number of obstacle collision:** Number of collisions during running tests, including dynamic and static obstacles.
- **Minimum/maximum allowable width:** The minimum allowable width refers to the minimum passing distance without collision. The maximum allowable width means that no collision will occur through roadways that exceed this width.
- **Temporal coefficient:** Spatial coefficient is the ratio of the actual time to the minimum travel time, representing the time of planner execution.
- **Spatial coefficient:** Spatial coefficient is the ratio of the actual distance traveled to the planned distance, representing the efficiency of local planner execution.
- **Smooth coefficient:** Smooth coefficient is the ratio of the actual Angle turned to the accumulated Angle of the planned path, representing the curvature of the path executed by the local planner.

#### 4.4 Experiment Results

To evaluate the moiton planning framework on our test vehicle described in Sect. 3, the combination algorithms are run based on the simulation mining robot tasks in Sect. 4.2. We test our framework used the mentioned metrics in Table 1, the running state of the test is shown as Figs. 5, 6, 7 and 8.

As shown in the Table 2, all of the global planner can gain connected path. RRT\* has the highest planning efficiency, with planning time of 230.1 ms, while Hybrid A\* is tens of times longer than RRT\*, that of 5957.2 ms. In the same planning task, the performance of the four algorithms is relatively close, so the global path searched by all used algorithms is relatively close to the optimal path in the coal mine roadway. A\* and RRT\* can more suitable for global path planning.

Table 3 shows that in the static obstacle avoidance test, DWA has more collisions than Trajectory Rollout. As the DWA algorithm is more aggressive in plan, it can get through narrower tunnels. The minimum passable distance of DWA is 80 cm, but DWA needs a wider safety pass width to ensure stable and safe passage. To meet the safety requirement of mine, mining robots are forbidden to collide and fall down the safety supports to the ground. However, there are lots of hydraulic props in mines. So the operation of the mining robot must ensure that it can not hit the safety support. The robust Trajectory Rollout algorithm is more suitable for the complex down-hole environment.

**Table 2.** Comparison of global planning

Gobal planner	Success rate	Path length	Spatial coefficient
A*	100%	289.9 ms	1.02
Dijkstra	100%	306.3 ms	1.03
RRT*	100%	230.1 ms	1.02
Hybrid A*	100%	5957.2 ms	1.04

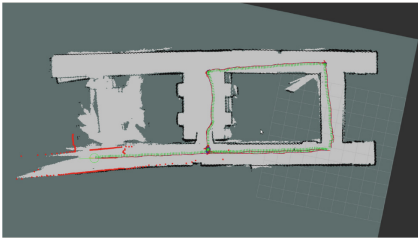


Fig. 5. Dijkstra of DWA

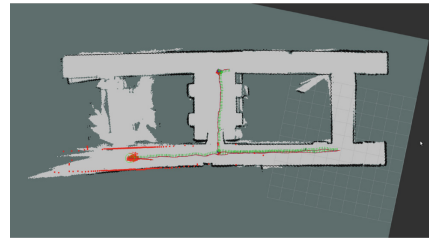


Fig. 6. A\* of DWA

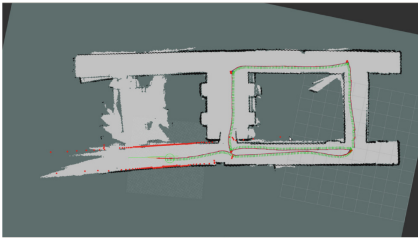


Fig. 7. RRT\* of Trajectory Rollout

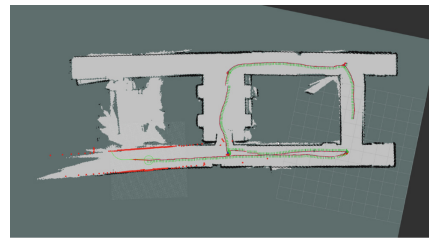


Fig. 8. Hybrid A\* of Trajectory Rollout

Table 3. Comparison of local planning

Gobal planner	Success rate	Recovery behavior	Static obstacle collision	Dynamic obstacle Collision	Minimum allow width	Maximum allow width
Trajectory rollout	98.3%	4	1	0	86 cm	86 cm
DWA	85.8%	2	6	0	80 cm	95 cm

Table 4. Comparison of assembly planner

Gobal planner	Local planner	Runtime	Temporal coefficient	Travel distance	Spatial coefficient	Smooth coefficient
Dijkstra	Trajectory rollout	9.3 min	5.44	76.39 m	1.49	1.67
A*		9.1 min	5.41	<b>69.46 m</b>	<b>1.38</b>	<b>1.41</b>
RRT*		10.7 min	6.26	74.73 m	1.46	5.32
Hybrid A*		<b>8 min</b>	<b>4.70</b>	72.34 m	1.42	1.80
Dijkstra	DWA	8.6 min	4.99	94.66 m	2.25	2.05
A*		7.5min	4.37	72.07 m	1.40	1.65
RRT*		<b>6.1 min</b>	<b>3.55</b>	67.77 m	1.31	1.74
Hybrid A*		12.2 min	6.98	<b>62.03 m</b>	<b>1.18</b>	<b>1.63</b>

Table 4 is a comparison of assembly planner, runtime, travel distance, and path smoothness. The global path generated by RRT\* and Hybrid A\* is the most suitable for the local planner of DWA. The running time of RRT\* of DWA is the shortest, which is 6.1 min, indicating RRT\* of DWA has the highest planning efficiency. The running distance and smoothness indexes of Hybrid A\* of DWA are the best, indicating that DWA can best follow the global path of Hybrid A\*. The travel distance of Hybrid A\* of DWA is the shortest, which is 62.03 m. However, Hybrid A\* of DWA has the longest running time to reach each task point due to excessive constraints and long operation time. In terms of running distance results, little difference is founded between the two assembly planners but RRT\* of DWA is considerably faster than Hybrid A\* of DWA. So RRT\* of DWA is the best among the four DWA combination algorithms tested.

The most suitable global path for Trajectory Rollout is generated by A\* and Hybrid A\*. A\* of Trajectory Rollout has the shortest running distance (69.46 m), and its running time is second only to that of Hybrid A\* of Trajectory Rollout. Hybrid A\* of Trajectory Rollout has the shortest running time, which is 8 min, and the shortest running distance after A\* of Trajectory Rollout. This indicates that Trajectory Rollout is very suitable for A\* and its improved algorithm.

As the shortest runtimes of the two groups, Hybrid A\* of Trajectory Rollout has a better perform than RRT\* of DWA. Meanwhile, Hybrid A\* of DWA's path is shorter than the A\*-Trajectory Rollout's path. Among the measured algorithms, RRT\* of DWA's running time has obvious advantages compared with other algorithms, which is suitable for the requirements of fast passage tasks such as underground rescue. Additionally, Hybrid A\* of DWA has the shortest distance and better smoothness, but the execution time is the longest, indicating that it is suitable for the application of narrow tunneled traffic and other scenarios that need to strictly follow the route.

## 5 Conclusion

This paper set a mining robot motion testing framework for simulated conditions of the coal mine, including roadway following test, static obstacle avoidance test, and dynamic obstacle avoidance test. This paper using a differential wheeled robot platform to test the effect of the assembly planner. Several indicators, planning speed, travel distance, and trajectory smoothness have been evaluated. This paper provides some guidelines for the application of coal mine robots.

In the study findings, it is clear that RRT\* is the fastest planning global path. Additionally, RRT\* of DWA can reach the target point faster than other algorithm combinations in the simulated mining environment. It is suitable for fast arrival scenarios, such as rescue and transport missions. Due to considering kinematic constraints, the planning time of Hybrid A\* is much longer than that of other algorithms, but the path by Hybrid A\* is the easiest to execute. Among various algorithm combinations, Hybrid A\* of DWA algorithm has the shortest travel distance and the smoothest path, indicating that it is suitable for slow but strict path implementation scenes, such as narrow roadways with many obstacles.

## References

1. Wang, G., Liu, F., Pang, Y.: Coal mine intellectualization: the core technology of high quality development. *J. China Coal Soc.* **44**(2), 349–357 (2019)
2. Ge, S., Hu, E., Pei, W.: Classification system and key technology of coal mine robot. *J. China Coal Soc.* **45**(1), 455–463 (2020)
3. Marder, E., Berger, E., Foote, T., Gerkey, B., Konolige, K.: The office marathon: robust navigation in an indoor office environment. In: *IEEE International Conference on Robotics & Automation*, pp. 300–307 (2010)
4. Li, X., Sun, Z., Cao, D., et al.: Real-time trajectory planning for autonomous urban driving: framework, algorithms, and verifications. *IEEE/ASME Trans. Mechatron.* **21**, 740–753 (2015)
5. Luecken, M., Büttner, M., Chaichoompu, K.: Benchmarking atlas-level data integration in single-cell genomics. *Nat. Methods* **19**, 41–50 (2022)
6. Shivam, J., Soumya, S.: Low-cost path planning in 2D environment using A\* algorithm by considering slope of the obstacle. *IFAC-Papers OnLine* **55**(1), 783–788 (2022)
7. Richards, N., Sharma, M., Ward, D.: A hybrid A\*/automaton approach to on-line path planning with obstacle avoidance. In: *AIAA 1st Intelligent Systems Technical Conference*, vol. 19, pp. 9–10 (2004)
8. González, R., Jayakumar, P., Iagnemma, K.: Stochastic mobility prediction of ground vehicles over large spatial regions: a geostatistical approach. *Auton. Robots* **41**(2), 311–331 (2017)
9. Qin, Y.: Research on detection system for coal mine detection robot based on the technology of information fusion. In: *International Technology and Innovation Conference*, pp. 151–152 (2009)
10. Palmieri, L., Kai, O.A.: A novel RRT extend function for efficient and smooth mobile robot motion planning. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 205–211 (2014)
11. Tian, Z., Gao, X.: Path planning of disaster relief robot based on improved artificial potential field method. *Ind. Mining Autom.* **42**(09), 37–42 (2016)
12. Lei, Y., Wang, Y., Wu, S., Gu, X., Qin X.: A fuzzy logic-based adaptive dynamic window approach for path planning of automated driving Mining Truck. In: *2021 IEEE International Conference on Mechatronics*, pp. 1–6 (2021)
13. Roesmann, C., Feiten, W., Woesch, T.: Trajectory modification considering dynamic constraints of autonomous robots. In: *Robotics Proceedings of ROBOTIK 2012 7th German Conference on VDE*, pp. 1–6 (2012)
14. Steve, M.: The marathon 2: a navigation system. In: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2718–2725 (2020)
15. Lu, D., David, V., Dave, H., William, D.: Layered costmaps for context-sensitive navigation. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 709–715 (2014)