Research on Planar Path Planning of Mobile Robot Based on Cellular Automata

Wenwei Li[12*], Weixi Huang[2], Yongqian Xu[23], Jianan Liang[23], Hao Qin[23], Chong Zhang[234]

[1] Foshan Zhiyouren Technology Co., Ltd., Foshan, China

[2] South China Robotics Innovation Research Institute, Foshan, China

[3] Intelligent Manufacturing Institute, Guangdong Academy of Sciences, Guangzhou, China

[4] Wuyi University, Jiangmen, China

* liwenwei011@163.com

https://doi.org/10.70695/10.70695/IAAI202503A14

Abstract

This study designed planar path planning algorithm for mobile robots based on Cellular Automata (CA). The construction of the algorithm was expounded including the cellular space, cellular states, neighborhood and evolutionary rule. The distance from each free cell to the target point was generated through the evolution of CA. Then, the negative gradient tracking mechanism was introduced to generate the global optimal path based on the cellular state values. To enhance real-time performance, the time complexity of the algorithm was reduced by stack-based cellular states updating strategy. Verified by experiments on the Matlab simulation and physical robots, this algorithm demonstrates a stable ability to generate the shortest path in complex obstacle environments.

Keywords Path Planning; Mobile Robot; Cellular Automata

1 Introduction

Path planning in dynamic environments faces huge challenges, especially in terms of meeting the requirements of environmental constraints and kinematics performance. Mobile robot path planning refers to the design of the safely collision-free path with shortest distance and least time-consuming from the starting point to the end point by a mobile robot autonomously [1]. Generally speaking, the path planning can be divided into two categories: the global path planning and the local path planning, according to whether all the information of the environment is accessible or not [2]. Nowadays, bio-inspired techniques had been successfully applied to robot's path planning problems. Among these technologies, CA models provide effective means for guiding robots through static and dynamic environments, adeptly navigating obstacles to reach desired destinations [3].

CA is a dynamic system composed of discrete finite cells, which evolves in discrete time according to the same evolutionary rules, so as to simulate the behaviors of complex systems. When the global environment is known, CA can be used for global shortest path planning of mobile robot. The CA consists of four parts: cellular space, cellular states, neighbors and evolutionary rules. The mathematical form of standard CA is a quadruple:

$$A = (Ld, S, N, f) \qquad (1)$$

Where, A represents a CA model; Ld represents a d-dimensional cellular space, $d \in Z+$; S is a finite set of discrete states of a cell; N represents the neighbors associated to the central cell; f is the evolutionary rules between the central cell and its neighbors.

These years, some representative studies had emerged on the path planning by using CA [4-11]. Zeng M, et al [4] proposed a kind of the path planning algorithm based on CA. Accordingly, a cellular state evolution rule was designed, and the optimal path search method was determined according to the evolution of cellular state. Syed UA, et al [5] proposed an efficient path-planning scheme based on CA that generates optimal paths in the minimum time, and performance comparisons with A*, Dijkstra, D* and MPCNN have proven it to be time-efficient. Santoso J, et al [6] proposed a new approach to path planning for mobile robots with cellular automata and cellular learning automata, and divided the path planning into two stages, namely the global path planning based on CA and the local path planning based on cellular learning automata. Sedreh Z, et al [7] presented a method for solving robot routing problem using CA and genetic algorithm(GA). In this method, the working space model and the objective function calculation are defined by CA, and the generation of initial responses and acceptable responses is done using the GA. Li W, et al [8] analysed the picking performance of a robotic mobile fulfillment system (RMFS) and proposed a Simulation Framework of RMFS based on Cellular Automata (SFRMFSCA). Duan Y, et al [9] proposed the Cellular Automata Slime Mold Algorithm (CASMA), which enhanced slime mold algorithm accelerates convergence speed and improves search accuracy. Zhu F [10] studied the problem of full coverage path planning for mobile robots based on Biological Inspired Neural Network Cellular Automata (BINN-CA) system. The BINN algorithm is used for full coverage path planning, and the escape mechanism when the robot falls into the dead zone is designed based on CA. Our team [11] had previously utilized CA to plan the obstacle avoidance path for mobile robots and achieved good results. The above-mentioned researches theoretically confirmed the feasibility of using CA algorithm for path planning of mobile robots. However, due to the lack of research on reducing the time complexity of the algorithm, the proposed algorithm was rather time-consuming to calculate when applied in engineering practice.

The novelty of this study lies in the proposal of a universal method for generating the shortest path of mobile robots using CA in any environment, and the presentation of a stack-based cells states update strategy to reduce the time complexity of the algorithm. In the following parts, we will use CA to carry out global path planning for mobile robot, and elaborate its principle in essence.

2 Design of Cellular Automata

2.1 Cellular Space

Creating a rectangle space L2, which composited of m×n square cells. Row label are denoted by i (1 ≤ i ≤ m) and column label by j (1 ≤ j ≤ n), namely:

$$L_2 = \bigcup S_{i,j} \left(i = 1,\cdots,m\,;\, j = 1,\cdots,n\right) \tag{2}$$

2.2 Cellular States

In the initial states of CA, each cell's state is defined as Table 1.

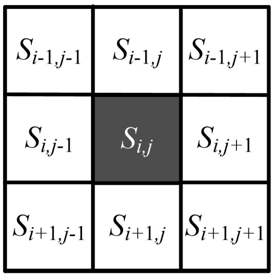
Table 1. The value of cellular state

Cellular state	Meaning
0	free cell
1	obstacle cell
2	starting cell
3	ending cell

2.3 Neighbors Definition

The neighbors definition of two-dimensional CA is more complex, including Von. Neumann type, Moore type and other forms [12]. Since the mobile robot can move in any directions, Moore type neighbors are adopted. As shown in Fig. 1, Moore type neighbors for central cell Si,j is defined as:

$$\begin{aligned} N_{\text{moore}}\left(S_{i,j}\right) = \{ & S_{i-1,j-1}, S_{i-1,j}, S_{i-1,j+1}, S_{i,j-1}, \\ & S_{i,j+1}, S_{i+1,j-1}, S_{i+1,j}, S_{i+1,j+1} \} \end{aligned} \tag{3}$$



$S_{i\text{-}1,j\text{-}1}$	$S_{i\text{-}1,j}$	$S_{i\text{-}1,j+1}$
$S_{i,j\text{-}1}$	$S_{i,j}$	$S_{i,j+1}$
$S_{i+1,j\text{-}1}$	$S_{i+1,j}$	$S_{i+1,j+1}$

Fig. 1. Planar Moore type neighbors for cell $S_{i,j}$

2.4 Evolutionary Rules

The evolutionary rules of cell Si,j can be defined as:

$$S_{i,j}^{t+1} = f\left[S_{i,j}^{t}, N_{\text{moore}}\left(S_{i,j}^{t}\right)\right] = f\left(S_{i-1,j-1}^{t}, S_{i-1,j}^{t}, S_{i-1,j+1}^{t}, S_{i,j-1}^{t}, S_{i,j}^{t}, S_{i,j+1}^{t}, S_{i+1,j-1}^{t}, S_{i+1,j}^{t}, S_{i+1,j+1}^{t}\right) \quad (4)$$

To mark the relative distance from each cell to the ending cell, the evolution rules of the cellular state is designed as:

When the value of current cell is 0: finding a cell among current cell's neighbors, and this founded cell has the smallest value in all the neighbors whose value is greater than or equal to 3. Then set the value of current cell equal to the value of this founded cell plus 1 or 1.4. Specifically, the diagonal neighbor plus by 1.4, and the non-diagonal neighbor plus by 1 to represent the cost of the relative position moving.

When the value of current cell is not 0: keeping the value of current cell unchanged.

Based on the above rules, the evolutionary example of a CA is shown in Fig. 2.

0	0	1	0	0	0	0	0
0	0	1	1	1	1	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	3

(a) Initial setting states ($t = 0$)

0	0	1	0	0	0	0	0
0	0	1	1	1	1	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	4.4	4
0	0	0	0	0	0	4	3

(b) States after first-step evolution ($t = 1$)

0	0	1	0	0	7.8	7.4	7
0	0	1	1	1	1	6.4	6
0	0	0	7.8	6.8	5.8	5.4	5
0	0	0	7.4	6.4	5.4	4.4	4
0	0	0	7	6	5	4	3

(c) States after fourth-step evolution ($t = 4$)

11.6	11.2	1	9.8	8.8	7.8	7.4	7
11.2	10.2	1	1	1	1	6.4	6
10.8	9.8	8.8	7.8	6.8	5.8	5.4	5
10.4	9.4	8.4	7.4	6.4	5.4	4.4	4
10	9	8	7	6	5	4	3

(d) States after seventh-step evolution ($t = 7$)

Fig. 2. The evolutionary example of a CA

3 Generation of Path

CA model is evolved according to above rules until all cell values are not zero. Based on the state of the cells, it is easy to search for the shortest path by using the greedy strategy. The shortest path from the starting point to the ending point is planned according to the idea of the negative gradient. The negative gradient direction is the direction where the function decreases fastest, and the shortest path can be obtained as long as the mobile robot moves along the negative gradient direction. The negative gradient in two-dimensional space is generally presented as:

$$-grad\ g(x,y) = -\nabla g(x,y) = -\frac{\partial g}{\partial x}\boldsymbol{i} - \frac{\partial g}{\partial y}\boldsymbol{j} \tag{5}$$

The specific method is that taking the starting cell as the robot's first position, and then finding a cell among neighbors of current cell as the next position, and requires that this founded cell has the smallest value among all the neighbors with a value greater than 2. Repeating the above process until the robot reaches the ending point. The mathematical expression for finding the next position $[i, j]^{\mathrm{T}}$ based on the current position $[i', j']^{\mathrm{T}}$ is as follows:

$$\begin{cases} [i,j]^{\mathrm{T}} = \underset{i,j}{\arg\min}\ N_{\text{moore}}\left(S_{i',j'}\right) \\ \text{s.t. } N_{\text{moore}}\left(S_{i',j'}\right) > 2 \end{cases} \tag{6}$$

The optimum path is obtained by connecting the adjacent points passed by the robot with a straight line. The flowchart of generating the shortest path is shown in Fig. 3.

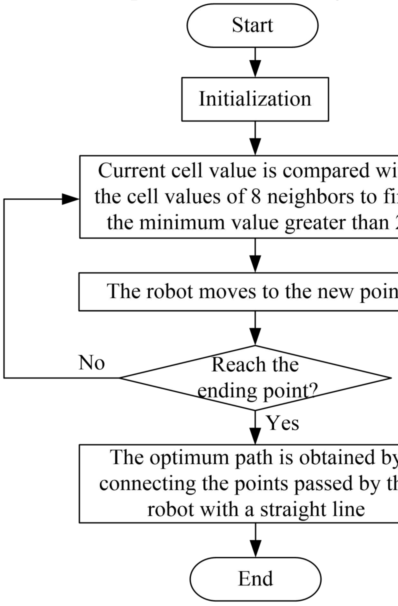
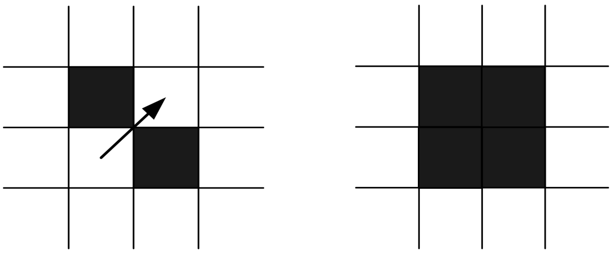


Fig. 3. Flowchart for generating the shortest path

Sometimes, the phenomenon of passing across obstacles occurs as Fig. 4(a). The solution is to artificially fill the diagonal paired free cells in advance as Fig. 4(b).



(a) passing across obstacles (b) after filling the diagonal paired free cells

Fig. 4. The phenomenon of passing across obstacles and pretreatment strategy

The search algorithm of the diagonal paired free cells is shown in Fig. 5. Creating a 2×2 sliding window that starts the search from the top left corner until it reaches the bottom right corner. The window moves one grid at a time, and when a row is completed searched, the window moves down one row and continues searching the next row. When the diagonal paired free cells are found, setting their state values to 1. After the above pretreatment, the phenomenon of passing across the obstacles will no longer occur.

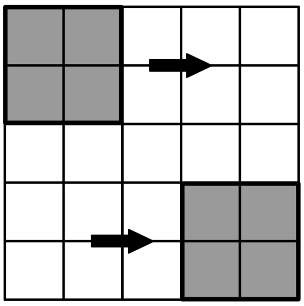


Fig. 5. The search algorithm for the diagonal paired free cells

4 Reduction of Time Complexity

Generally speaking, CA algorithm needs to go through multiple rounds of iterations to reach a stable state. In each evolution, the computational load for updating the cellular states is relatively large. Especially in large-scale high-resolution simulations, the computational load increases exponentially. The time complexity of our planar path planning algorithm based on CA is:

$$T(n) = O(k \times c \times t) \qquad (7)$$

Where, T represents the time complexity of the algorithm; k represents the number of neighbors of a cell; c represents the total number of free cells; t represents the iteration times of CA.

Although our previous research had achieved CA-based path planning [11], but the time complexity of the algorithm is relatively high, which restricted the engineering application of the algorithm. The complexity of the algorithm mainly stems from the repetitive calculation of cellular states. To avoid this situation, the data structure of three segments stack was applied. Specifically, the stack was divided into three segments, which stored zero cells with changed neighbor, zero cells with unchanged neighbor, and non-zero cells, respectively. Schematic diagram of cell popping and pushing into the stack is shown in Fig. 6.

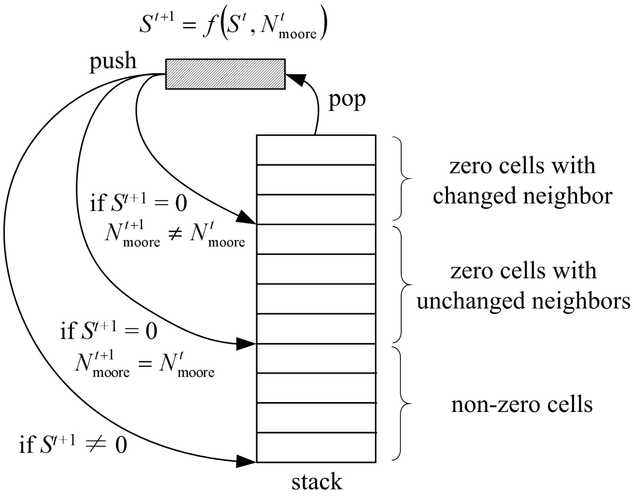


Fig. 6. Schematic diagram of cell popping and pushing onto the stack

For each evolutionary step, a cell is poped from the top of the stack for calculation, and the position where the cell is pushed into the stack is determined based on the calculation result. The flowchart of above process is shown in Fig. 7.

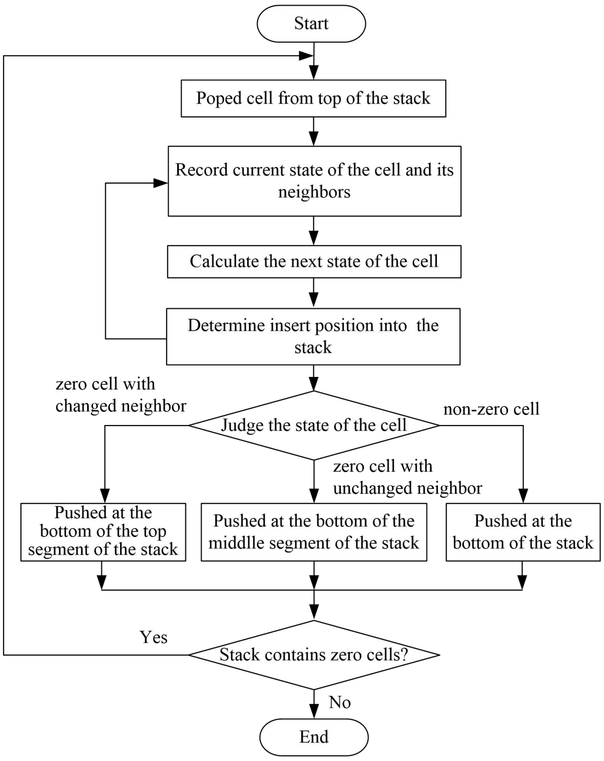


Fig. 7. Flowchart of stack-based cellular states calculation

Here, three situations will be considered. If the state of the cell is non-zero after calculation, it is pushed at the bottom of the stack. If the state of the cell is zero and neighbor changed after the calculation, it is pushed at the bottom of the top segment of the stack; If the state of the cell is zero and neighbor unchanged after the calculation, it is pushed at the bottom of the middle segment of the stack. Repeating the above process until all cells' states in the stack are non-zero value.

By applying the above method, the time complexity of the algorithm becomes:

$$T(n) = O(k \times c' \times t) \tag{8}$$

Where, c' represents the total number of free cells with changed neighbor, $c' < c$.

5 Simulation and Experimental Verification

As an example, an indoor environment with many rooms and corridors is defined arbitrarily, and its aerial view is shown in Fig. 8. The simulation is carried out in a 50×50 square space. Each square grid is greater than the minimum enveloping square of the robot's outline. The starting point of the robot is set at the top-left cell (1, 1), and the ending point is set at the bottom-right cell (50, 50).

First of all, we pretreated the original grid map according to the aforementioned method to eliminate the diagonal paired free cells. The original map and the processed map are respectively shown in Fig. 8(a) and 8(b).

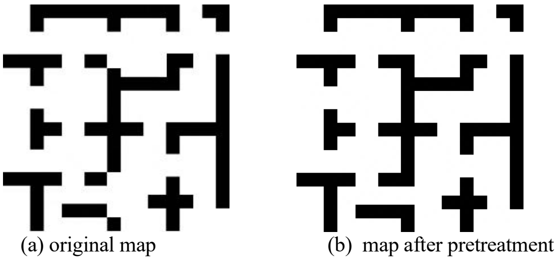

(a) original map (b) map after pretreatment

Fig. 8. Grid map definition and pretreatment

Generating a path involves two stages. For the first stage, based on the cellular evolutionary rule, CA evolved from the initial states to the non-zero final states. For the second stage, the obstacles avoidance path was generated by the idea of the negative gradient, and the found path is shown as the red line in Fig. 9. After comparison, we found that the path is basically consistent with the path planned using the Dijkstra algorithm. Here, we assumed that the robot is moving grid by grid, so, the path consists of a combination of horizontal lines, vertical lines, and 45° oblique lines. As a contrast, the physical experiment is carried out by using real mobile robot to make it travel on the path planned by our CA algorithm, and several on-site pictures is shown in Fig. 10.

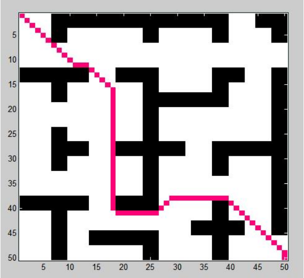
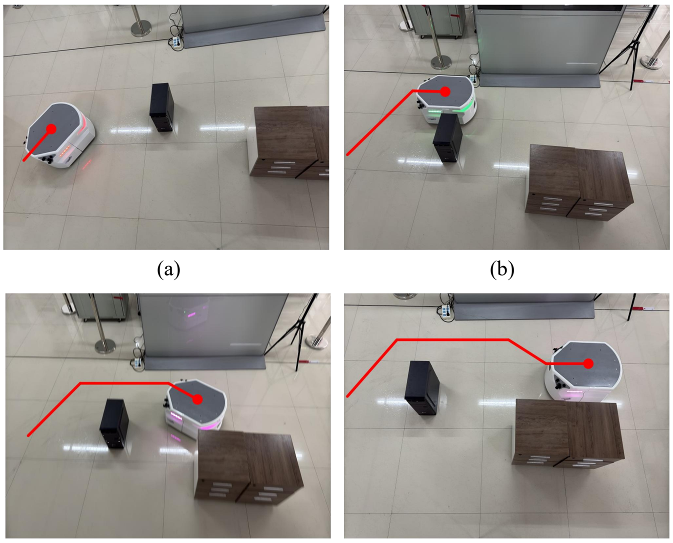


Fig. 9. Planar path planned by the CA algorithm



(a) (b)

(c) (d)

Fig. 10. The experiment of mobile robot running on the path planned by the CA algorithm

It should be emphasized that the actual running path (red line in Fig. 10 of the mobile robot is obtained by offsetting the theoretical path (green line in Fig. 9) based on the mobile robot's safety rotational radius. From Fig. 9 and Fig. 10, we can see that the CA algorithm can get the shortest path when the global environment is known, which shows that the CA algorithm has excellent ability of path planning.

6 Conclusions

This paper proposed a algorithm for planar path planning of mobile robots based on CA. The contributions and innovations of this paper are as follows: (1) a square grid is used to disperse the two-dimensional environmental space, and each grid corresponds to a cell, and free cell's state is presented as the distance between free cell to the ending cell; (2) central cell's evolutionary rules based on its Moore type neighbors' states is proposed, which made CA evolves from an initial states to non-zero final states; (3) a pretreatment strategy is presented to avoid the phenomenon of passing across obstacles by searching and filling the diagonal free paired cells; (4) By using a stack data structure to store cellular states, only cells with a state of zero and its neighbor's states has changed are considered to calculated, while non-zero cells are not recalculated, which significantly reducing the computational complexity of our algorithm; (5) the idea of negative gradient is borrowed to decide where is the next position to move, and the shortest path search algorithm is designed. Simulation and physical experiment showed that the

CA algorithm is suitable for the shortest path planning when the global environment is known, and has the characteristics of simple arithmetic calculation, fast speed and high efficiency. In future work, we will continue to study CA algorithms that integrate other intelligent algorithms and heuristic information to achieve more intelligent and faster planar path planning.

Acknowledgement

This work was supported by Foshan Science and Technology Innovation Project under Grant FS0AA-KJ919-4402-0060, and Core Technologies Research Project of Shunde District under Grant 2130218003022.

Conflicts of Interest

The authors declare no conflicts of interest.

References

1. Liu L, Wang X, Yang X, et al. Path planning techniques for mobile robots: review and prospect[J]. Expert Systems with Applications, 2023, 227: 120254, .
2. Zhang H, Lin W, Chen A. Path planning for the mobile robot: a review[J]. Symmetry, 2018, 10(10): 450 .
3. Souza NLB, Lima DA. Systematic literature review on the application of controllers based on cellular automata in robotic tasks[J]. Revista de Informática Teórica e Aplicada, 2024, 31(2): 120-137.
4. Zeng M, Wang C, Liu G, et al. Mobile robot path planning based on cellular automata[J]. Journal of Nanchang University (Engineering & Technology), 2012, 34(3): 287-290.
5. Syed UA, Kunwar F. Cellular automata based real-time path-planning for mobile robots[J]. International Journal of Advanced Robotic Systems, 2014, 11(7): 93.
6. Santoso J, Riyanto B, Adiprawita W. Dynamic path planning for mobile robots with cellular learning automata[J]. Journal of ICT Research & Applications, 2016, 10(1).
7. Sedreh Z, Sadeghzadeh M. Robot path planning using cellular automata and genetic algorithm[J]. Journal of Advances in Computer Engineering and Technology, 2019, 5(1): 19-26 .
8. Li W, Miao L, Yang P. Simulation analysis of robotic mobile fulfillment system based on cellular automata[J]. International Journal of Simulation Modelling, 2021, 20(3): 583-594.
9. Duan Y, Yang Y, Cao Y, et al. Path planning optimization for swine manure‐cleaning robots through enhanced slime mold algorithm with cellular automata[J]. Animal Science Journal, 2024, 95(1): e13992.
10. Zhu F. Mobile robot complete converge path planning based on BINN-cellular automata system[J]. Machinery Design & Manufacture, 2024, 8: 346-349.
11. Li L, Li W, Huang J. Obstacles avoidance path planning for mobile robot based on cellular automata[C]. 2024 6th International Conference on Robotics, Intelligent Control and Artificial Intelligence (RICAI), IEEE, 2024: 60-64.
12. Rollier M, Zielinski KMC, Daly AJ, et al. A comprehensive taxonomy of cellular automata[J] Communications in Nonlinear Science and Numerical Simulation, 2025, 140: 108362.

Biographies

1. **Wenwei Li** PhD, Senior Engineer, with main majors and research directions in intelligent robotics, CAD/CAE based digital simulation, mechanical dynamics, and manufacturing informatization. He has applied for 55 technical patents and published over 30 academic papers.
2. **Weixi Huang** Master degree, engineer, major and research directions: automation technology, optical intelligent processing and testing equipment technology. He has applied for more than 30 technical patents and published over 10 academic papers.
3. **Yongqian Xu** Master degree, professorate senior engineer, main research directions: automation technology and mechatronics professional field, he has applied for technical patents, published more than 20 academic papers, and obtained more than 20 technical patents.
4. **Jianan Liang** PhD, senior engineer, main research areas: automation technology, machine vision, and intelligent manufacturing technology. He published over 30 academic papers and applied for over 40 technical patents.

5. **Hao Qin** PhD, Senior Engineer, main research directions: knowledge graphs, intelligent design, digital twins. he has published more than 10 SCI/EI indexed academic papers and applied for over 40 technical patents.
6. **Chong Zhang** Master's degree student, main research areas: machine vision, intelligent detection and intelligent control technology. He published 3 academic papers and applied for 1 technical patent.

基於元胞自動機的移動機器人平面路徑規劃研究

李文威[12]，黃偉溪[2]，徐永謙[23]，梁佳楠[23]，秦昊[23]，張衝[234]

[1]佛山智優人科技有限公司，佛山，中國，528300

[2]華南智能機器人創新研究院，佛山，中國，528300

[3]廣東省科學院智能制造研究所，廣州，中國，510070

[4]五邑大學，江門，中國，529020

摘要：設計了一種基於元胞自動機的移動機器人平面路徑規劃算法。從元胞空間、元胞狀態、鄰域和進化規律等方面闡述了該算法的結構。通過CA進化生成每個自由細胞到目標點的距離，然後引入負梯度機制，根據細胞狀態值生成全局最優路徑。為了提高算法的實時性，採用了基於棧的元胞狀態更新策略來降低算法的時間復雜度。通過Matlab仿真和實體機器人實驗驗證，該算法在復雜障礙物環境下具有穩定的最短路徑生成能力。

關鍵詞：路徑規劃；移動機器人；元胞自動機

---

1. 李文威，博士，高級工程師，主修專業及研究方向為智能機器人、CAD/CAE數字仿真、機械動力學及製造信息化。他已申請55項技術專利，發表30多篇學術論文；
2. 黃偉溪，碩士，工程師，專業及研究方向：自動化技術、光智能處理與檢測設備技術。他已申請30多項技術專利，發表10多篇學術論文；
3. 徐永謙，碩士，正高級工程師，主要研究方向：自動化技術及機電專業領域，他已申請技術專利，發表20多篇學術論文，獲得20多項技術專利；
4. 梁佳楠，博士，高級工程師，主要研究領域：自動化技術、機器視覺及智能製造技術。他發表30多篇學術論文，申請40多項技術專利；
5. 秦昊，博士，高級工程師，主要研究領域：知識圖譜、智能設計、數字孿生。他發表了SCI/EI論文10多篇，申請/授權技術專利40多項；
6. 張衝，碩士研究生，主要研究領域：機器視覺、智能檢測與智能控制技術。他發表3篇學術論文，申請1項技術專利。