Micro Mouse Maze Solving

*Ching-Hsing Pei **Jing Lee

*Center for General Education, Tung Fang Design University,

**Department of Electronics Engineering, Southern Taiwan University

Abstract

Micro mouse is a device that applied the principles of optical, mechanical, electronic, and integration of hardware and software technology. This paper discusses the micro mouse maze search method and describes the structure of information of the maze; and then using graph theory defines a maze search problem. To achieve the purpose of maze search, the paper pointed out that the error of the depth first search and its correction by a superior algorithm.

Keywords: Micro mouse, maze, depth first search

摘要

電腦鼠是一種應用光學的原理，機械，電子等集成的硬體和軟體技術的裝置。本文論述了電腦鼠標迷宮搜尋方法，並介紹了迷宮的結構信息;然後利用圖論定義迷宮搜尋問題。爲了實現迷宮搜尋的目的，文中指出深先搜尋法的缺點，並採用一個較好的算法來加以修正。

關鍵詞:電腦鼠，迷宮，深先搜尋

1.Introduction

IEEE Micro Mouse competition is held once a year, the international competition. A micro mouse is called a small automated wheeled robot must own in an unknown maze patrolling the maze to reach the center. The outcome of the micro mouse to reach the center by the time of decision [1-3]. The parts of software of micro mouse mainly to develop a maze search algorithm. In this article, detailed description of the wall following search and a variety of depth first search method, but also pointed out the shortcomings of depth first search, and use the flood fill method to amend this shortcoming. In IEEE competition, the wall following search usually can not be successfully completed to search the maze. Depth first search can be done though the maze search, but usually consuming too much time. The proposed amendment to the radial depth first search method is a better way usually.

2.Maze Structure and Graphics

Micro mouse maze search problem is letting micro mouse in the unknown state of the maze, searching the maze as soon as possible to reach the center of the maze. To better define this problem, the following will be on the entities of maze, data structure, the illustration of the graphical representation.

2.1 Maze Entities

Micro mouse maze is composed of 16 × 16 square cells, each square cell whose size is 18cm × 18cm, if the square with n walls is called a n wall cell (i.e. n <5). Maze wall height is 5cm, thickness is 1.2cm, wall color is white, the top is red, the coating materials of the wall and the top, it must be able to reflective a infrared projection. Maze on the ground is to the surface coated with a black gloss paints to the wood, paints must be able to absorb the infrared projection. Maze's "starting point" in the corner of the maze, "starting point" of the maze should be three of the walls. The "end" of maze is composed of four cells in the center, no walls between the four cells.

2.2 Structure of information of the maze

Each maze cell has four sides, whether four sides of the cell have walls or not can be expressed in a bit, so the walls of a maze cell position can be used of four bits NWSE to define the arrangement,

EAST = 1

SOUTH = 2

WEST = 4

NORTH = 8

Therefore, 1100 represent the cell have Northern and Western walls. Using this binary encoding, we can use 0-15 to define all the possible of the cell of the maze as Figure 1.

Figure 1. The maze cell’s coding

When the micro mouse maze search, it will using infrared sensors to detect the state of cell’s wall, when the infrared sensor is detected a cell’s wall, we must be added information of the wall to the cell's record, the program must be implemented at this time join the wall operation. Note that this operation should use bit operations in the '|' operator, do not use '+', otherwise it will cause an error. Give an example of the following: if we want to cell [2] [3] to join the north wall, the operator should be written as

cell [2] [3] = cell [2] [3] | NORTH

Paying particular attention to when we joined the wall of a cell, the adjacent cell must be added to the corresponding wall, for example, for cell [2] [3] to join the north wall, we must be cell [3] [3] to join the south wall, that is, cell [3] [3] = cell [3] [3] | SOUTH

2.3 The graphical representation of the maze

For a micro mouse maze, we can use a graphical structure to define it. Graph is the set of nodes and edges. In the maze, end point, starting point and in addition to two walls outside the cell are defined as nodes; nodes and nodes connected by edges, the definition of the edge length is of the distance between two end points of the edge. To facilitate the definition of side length of a cell as 1 unit, then the side length represents the cell number of the interval between two end points.

Figure 2 is a 10 × 10 maze, the starting point and the dead end alley with a black node, the blue node to represent bifurcation junction, the red node representing end of maze, sides to the green line, and number on side to represent the side length. We can find out to get a layout with many loops.

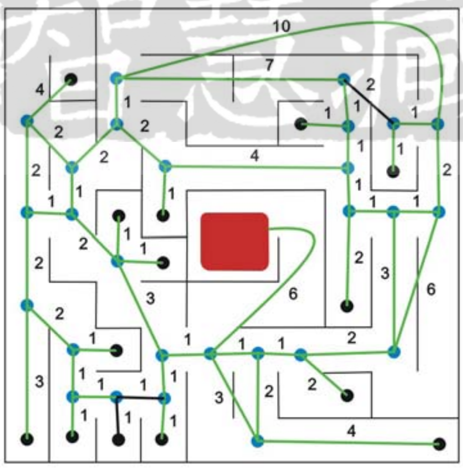

Figure 2. A example of the maze and their corresponding graphic

3.Solving Method of Maze Problem

Micro mouse commonly used search algorithm is wall following method, depth-first method and flood fill method.

3.1 Wall Following Method

Wall following method is an old and simple algorithm, the micro mouse in the maze entrance to choose one of the walls, after move along the wall until you reach the end date. The program of wall following method is very simple, but he did not guarantee to solve the maze search problem [5]. This is especially in the design of the maze of micro mice are usually designed to can not be answered to use wall following method.

3.2 The Depth First Search

Graph search method can be divided into the depth-first search method (Depth-First Search, DFS) and width-first search method (Width-First Search, WFS) [6]. The micro mouse in the maze problem, because of the micro mouse does not know in advance the state of the maze, it can only according to have traveled around the path and the current cell environment, and try to find the path to reach the end point. Therefore, depth-first search method is relatively straightforward and easy way. In the depth-first search method, when the micro mouse to face the node, and randomly selected a side that is not traversed. If all the sides have been traversed, the micro mouse returned back to the previous node, by depth-first principle to choose another one that did not passed side to continue down to go. In the node, according to the choice of different way of side, can be divided into the following methods:

1. The right hand rule: the priority right turn, followed by the straight, turn left.
2. The left hand rule: first left, then straight ahead, turn right.
3. The Mid-left rule: the priority straight, followed by the left, turn right;
4. The Mid-right rule: the priority straight, followed by the right, turn left;
5. The random rule: no fixed priority, the decision by random;

The above method will search each path in the graph, each connected node, until you reach the end date. Therefore, their advantage is as long as there exist a path to the end, it will certainly be able to find the answer.

3.3 Radial Depth First Search

Radial depth first search method is a modified depth first search method. This method in case of fork in the road, it will identify the current location relative to the location of the center, and choose the side of preferential access to the maze center. In this way just as tourists traveling in Taipei, if he wanted to 101Building, when he met fork in the road, he will look up the position of 101 Building, and then select the street for 101 Building advance. This approach is usually more effectively than randomly selected one road to move forward at the fork in the road. As the radial depth first search algorithm to make good use of the knowledge that end is in the center of the maze, so the search path length is usually shorter. Radial depth first search algorithm is usually the first to establish the table in Figure 3 of the weight, when the node encountered fork in the road priority to the smaller weight cell forward.

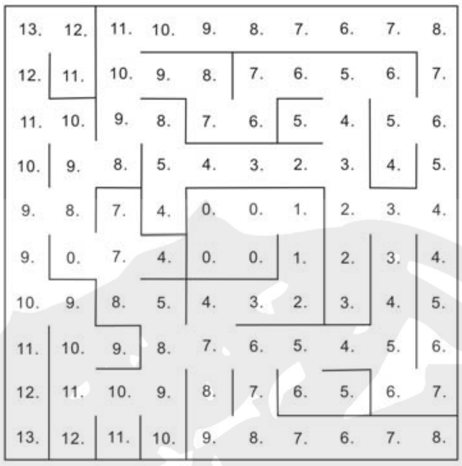

Figure 3. The table of the weight of the radial rule

Figure 4 is an example of the radial depth first search, the red path is the path traversed by the micro mouse. Using the radial depth first search algorithm, the micro mouse to memory nodes and the path traversed in order to avoid duplication of already traversed path. In Figure 5, the gray path represents dead-end alley, micro mouse should have the ability to remember these dead-end alleys in order to simplify the maze. Our practice is when the micro mouse into a 3-side cell (that is, the end of dead- end alley), it must go back to leave the cell, and left to join a virtual wall to leave the cell to make it become a closed cell and continue after closure of the side of the cell until you return to the previous node so far. Using this approach, we can correctly identify the dead-end alley of Figure 5.

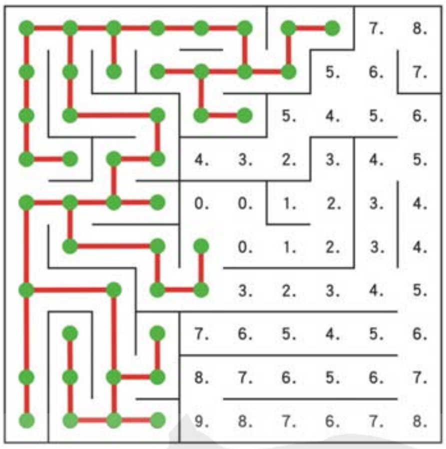

Figure 4. An example of the radial depth first search

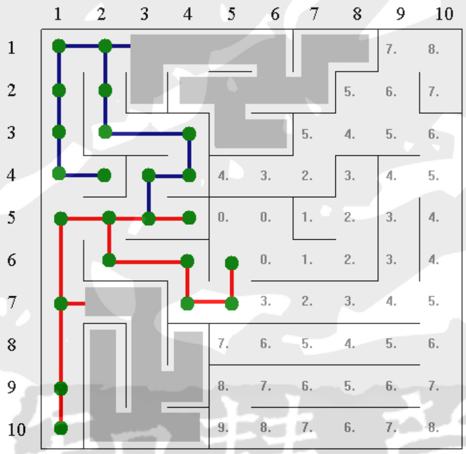

Figure 5. The maze and dead-end alley

3.4 The Amendment to the Radial Search

As described in previous sections, using depth first search method, when the micro mouse into the dead-end alley or the node have gone through all sides, the micro mouse must be returned back to the previous node. In Figure 5, for example, when the micro mouse along the blue path to go (4, 2) cell, it must be along the blue path back to the (5, 3) cell. However, it can be found here, in fact, when the micro mouse in (4, 1) cell, if down the road and turn right 2 steps to get to (5, 3) cell, with only a walking distance of the four cell than the original path to be 12 cell distance, it can be said to be greatly reduced. This is not the desired results, because of the traditional depth first search method does not take into account the length of the side, in fact, in the depth first search method, the side lengths are assumed to be 1. However, the micro mouse maze problems, the side lengths are not all one, so return the micro mouse on a node, in fact there shortest path problem.

This article uses the flood fill method [4] to solve this shortest path selection. Flood fill method can be thought of in a flood from the maze cell emission, then the flood would flow to a road where every movement of a cell on the cell to increase the number of code until you reach the objective point. Therefore, in Figure 5, we use (4, 1) cell for the flood emission points, as shown by the code in Figure 6.In Figure 6, we can be found (4, 1) back to last node of the shortest path length is 3, and you can find this path.

Although the flood fill method can be used to find the micro mouse back to the previous node of the shortest path, but we still need the program along the path back to the original node to achieve the depth first search of the original design, otherwise follow-up program caused the error. In other words, here take the path of the micro mouse is different from the actual depth first search path, but because of modern microcontroller operations faster than the speed of the micro mouse, it will not cause practical difficulties.

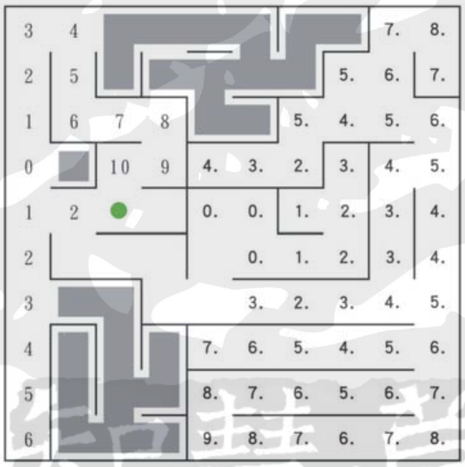

Figure 6. The coding of the flood fill method

The flowchart of the modified depth first search algorithm shown in Figure 7. Start the search from the bottom left corner of the micro mouse maze, encountered nodes moving to a small weight side. If they were all gone through fork road then to implement of the flood fill method to find the back of a node on the shortest path, the micro mouse along the shortest path arrived at a node, continue to use the above method of searching the maze, until you find the end date.

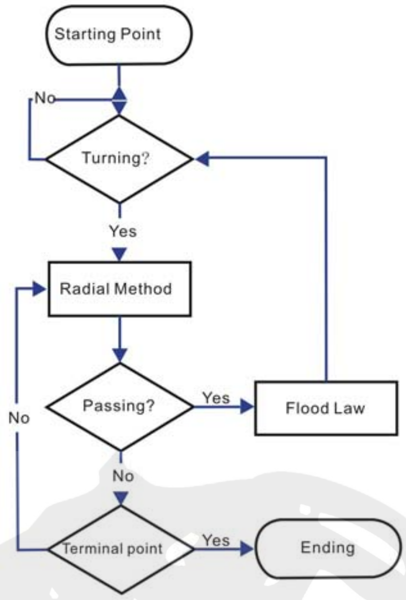

Figure 7. The flow chart of the maze search

3.5 Comparison and Analysis

We have used C language to write the above algorithm into visualization animation software. In this software, we can actually observe that the movement distance and the route of the micro mouse under the different algorithms, or even estimate the actual micro mouse movement time of needed to search a maze.

In Table 1, for example, we use 2004 and 2008 micro mouse competition maze in Japan [7] to compare with the classical radial depth first search algorithms and modified radial depth first search algorithms to the path length of search. It can be found in Table 1 compared to the traditional radial depth first search algorithm, the amendment radial depth first search algorithm does have significantly improved.

Further analysis as shown in Figure 8 of the 2008 micro mouse competition maze in Japan can be found, this maze has three loops to paint with a red line, these loops will allow the micro mouse in the depth first search, take a long detour back to the previous node. A modified radial depth first search algorithm can overcome this problem.

Table 1. Comparison of search steps of maze

Micro mouse race in Japan [7]	classical radial depth first search	Modified radial depth first search
2004	212	203
2008	358	326

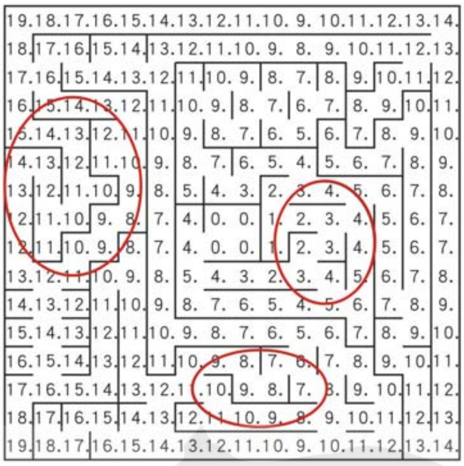

Figure 8. 2008 competition in Japan micro mouse maze [7]

4. Conclusion

Micro mouse maze search is an interesting question, though the history of the micro mouse has been nearly 60 years, so far no algorithm can be under state -- unknown maze, go directly from the starting point of the shortest path to reach the terminal; or even an algorithm can not ensure in any maze that better than other method always. This paper discusses the different forms of depth first search algorithm, we pointed out the errors of the depth-first search and developed a superior algorithm to modify these errors. Finally, we cited two examples to prove the validity of our method

5. References

1. M. Kassianidou, V. Srinivasan, and B. Villalobos, “Shannon’s work on AI,” in Claude Shannon and information theory, 1999. Available online: http://cse.stanford.edu/class/sophomore-college/projects-99/information-theory/ai.html
2. C. Christiansen, “Announcing the amazing micro mouse maze contest,” *IEEE Spectrum*, vol. 14, no. 5, p. 27, May. 1977.
3. R. Allan, “The amazing micromice: see how they won,” *IEEE Spectrum*, vol. 16, no. 9, pp. 62-65, Sep.1979.
4. 2009 Artificial intelligence robot single-chip micro mouse cum domestic and international tournament, http://robot2009.lhu.edu.tw/m1.htm
5. M. Sharma and K. Robeonics, “Algorithms for micro-mouse,”2009 International Conference on Future Computer and Communication, pp. 581-585, April 2009.
6. R. Tarjan, “Depth-first search and linear graph algorithms.” SIAM Journal of Computing, Vol. 1, No. 2, pp. 146-160, June 1972.
7. New Technology Foundation of Japan, http://www.ntf.org.jp