International Journal of Computer Science and Engineering (IJCSE)
ISSN(P): 2278-9960; ISSN(E): 2278-9979
Vol. 2, Issue 5, Nov 2013, 89-94

International Academy of Science, Engineering and Technology
Connecting Researchers; Nurturing Innovations

OPTIMIZATION OF FLOOD FILL ALGORITHM USING ITERATIVE LOOK-AHEAD AND DIRECTIONAL TECHNIQUE

VIPUL AGGARWAL
IP University, Delhi, India

ABSTRACT

In this paper further optimization of Flood fill and Flood-Ahead algorithm is proposed. Our aim is to reach the center of the maze with shortest distance and in fastest time possible. With the present work, a Micromouse traverses a maze of unknown dimensions by travelling the shortest path, without looking the path that exists beyond it, thereby increasing time complexity. Flood Ahead technique iteratively directs the Micro Mouse to the center of the maze by looking at cells (two) that exists beyond it. Time optimization is also achieved by not allowing the Micro mouse to travel those paths beyond which no further route(s) exist. In this paper, by using look-ahead algorithm iteratively with directional algorithm we further reduce the number of steps the Micro mouse has to travel. A performance comparison has been done with normal flood fill algorithm, line follower algorithm, flood-ahead algorithm and look ahead alone. The results show that it can significantly improve the performance which in turn will helps in improving the performance of flood-ahead algorithm.

KEYWORDS: Flood, Fill Algorithm, Iterative Flood, Ahead algorithm, Look-Ahead Algorithm, Directional Algorithm, IR Sensors

INTRODUCTION

A Micro mouse is a small microprocessor controlled vehicle that is able to navigate its way through an unknown and unconnected maze. It is a typical product of "mechatronics" embodying within itself an integration of computer and electronic technology and mechanics. There are several micro mouse competitions held in various countries. The objective for the contestant is to impart to the micro mouse an adaptive intelligence to explore different maze configurations and to work out the optimum route for the shortest travel time from start to finish.[11] A Micro mouse is a small microprocessor controlled vehicle that is able to navigate its way through an unknown and unconnected maze. It is a typical product of "mechatronics" embodying within itself an integration of computer and electronic technology and mechanics. There are several micro mouse competitions held in various countries. The objective for the contestant is to impart to the micro mouse an adaptive intelligence to explore different maze configurations and to work out the optimum route for the shortest travel time from start to finish. [11]

The micro mouse is a miniature robot designed to race inside a maze from a starting point to the destination. A maze is a tour puzzle in the form of a complex branching passage through which the solver must find a route. Along the way, there are walls and rooms with hedges that micro mouse has to go through and decide the shortest and fastest way the reach the destination. The destination is at the center of the maze. [7]

In order to make the micro mouse navigate properly to reach the destination inside the maze, it must be equipped with at least sensors, microcontrollers and motors. The most important equipment inside the micro mouse is controller that must be programmed with software to ensure micro mouse making wise decision and navigating smoothly.

What determines micro mouse is making wise decision using software algorithm inside the microcontroller. Developing algorithm for the micro mouse ultimately controls the autonomy positioning of the robot. [8] To accomplish this task, the micro mouse needs to map the maze through intelligent exploration, and then track the optimal (shortest) path which will allow the mouse to run in the shortest possible time. [7]

Micro Mouse applies the knowledge from different fields, such as computer science, computer engineering, electrical engineering, and mechanical engineering. It is a great opportunity to utilize interdisciplinary theoretical fundamentals to create a physical project. [7] The contributions to this paper are thus twofold. Firstly, to design and develop the optimizing techniques (time and distance) for Flood fill algorithm and Flood-Ahead algorithm and secondly to integrate Look Ahead technique and Directional algorithm with flood-fill algorithm as it will help to optimize the distance traveled and the time taken. This paper spans span across three major searching techniques, i.e. Look-Ahead technique, Directional Algorithm and Flood-Fill algorithm. The concept has been coined as **Iterative Flood-Ahead Algorithm.**

This paper is organized as follows. A brief introduction about the motivations for the research and development of Micro Mouse optimization is presented. In the next section, literature review of earlier proposed techniques is shown. The following section comprises of the framework that we have proposed for our entire Micro mouse system to optimize the time complexity. In last sections, the experiments conducted and subsequent performance evaluation is shown. Finally, at the end of this paper we conclude and give suggestions for future work in this field. [10]

RELATED WORK

Many researchers have contributed in the field of Artificial Intelligence optimizing the flood fill algorithm. The core algorithm was given by Lee [1]. After analyzing it, we saw that it was a Breadth First Search (BFS) algorithm; an application to Dijkstra's algorithm. But its drawback was that it suffered from huge memory requirements and slow performance. Many other authors have contributed in this area. In [2], the authors tried to improvise the Ackers [3] algorithm in which a two bit encoding technique was used per cell instead of saving the real distance. The authors proceeded in two phases: The filling phase: here they filled the neighboring cells to the starting point (S) with a one; the next cells were marked with two (these values are the distance of each cell from the source). They repeated this till they reached the target point (T). Retrace phase: If the target was reached in step I, they traced their way back by going to the cell with value I-1. They repeated this till they reached the starting point S. But this algorithm did not provide viable results for every maze. Hence, lead to its drawback. Consequently, an update procedure is used to modify the last search without re-starting. This technique is known as the Modified Flood Fill algorithm [4] [10], which again is widely used by micro mouse contesters. In [5], the authors used three approaches primarily. The wall follower logic: This worked on the rule of following either left wall or right wall continuously until it lead to the center. The micro mouse sensed the wall on the left or right, and followed it to wherever it leads until the center was reached. This simple logic used for solving the maze. Mathematical approach: Here, they had an array of 16 rows and 16 columns. If the array was denoted by M, then each cell was represented by M[R] [C]. They assumed the present position of array as M2 [R2] [C2], the previous position as M1 [R1] [C1], and the next position as M3 [R3] [C3]. [10] Now they proceeded as follows:

If R2-R1>0, then it is currently moving straight, upwards through the maze array.

If R2-R1<0, then it is currently moving straight, downwards through the maze array.

If C2-C1>0, then it is currently moving rightwards, through the maze array.

If C2-C1<0, then it is currently moving leftwards, through the maze array.

Depth-First Algorithm: Depth- First search was an intuitive algorithm for searching a maze in which the mouse first started moving forward and randomly chose a path when it came to an intersection. If that path leads to a dead end, the mouse returned to the intersection and chose another path. This algorithm forced the mouse to explore each possible path within the maze, and by exploring every cell, the mouse eventually found the center. It was called "depth first" [6] [10] because if the maze was considered a spanning tree, the full depth of one branch was searched before moving onto the next branch. The relative advantage of this method was that the micro mouse always found the route. Unfortunately the major drawback was that the mouse did not necessarily find the shortest or quickest route and it wasted too much time exploring the entire maze. [10]

Different researchers have tried in the area of optimization of various algorithms of micro mouse. But if their technique were simple, the time complexity of the algorithm was too large and if the time complexity was minimal, then the technique used was not practically possible. Till date the best algorithm to execute the program of micro mouse has been the flood fill algorithm. In our paper we have tried to further optimize the flood fill algorithm for better results. We have proposed a framework in which a cell looks ahead iteratively (for the cells with and without walls) every time it follows the flood-fill path. It also integrates Directional algorithm, which looks for the trans-versing neighbor cells and looks for the minimum distance towards center based on the value of the neighbor cells. It helps the Micro Mouse to eliminate those nodes where dead end exists and optimizes the distance travelled and the time taken.

PROPOSED APPROACH

In proposed framework I have combined Look Ahead algorithm and Directional algorithm with the existing Flood Fill algorithm to achieve optimization of distance and time. Modified version of Flood Fill algorithm helps us to put a constraint forward to check whether another variable can take a consistent value. In other words, it first extends the current partial solution with the tentative value for the considered variable. By combining the above two algorithms we aim to evaluate nodes ahead of the current node where Micro Mouse has to (will) travel. In our framework, nodes with walls (sensed through IR sensors) ahead of them will be eliminated and it will lead to optimum path utilization. [10] Iterative Flow-Ahead algorithm shows that how the micro mouse when kept at the starting of the maze reaches at the center using most optimum path and in minimum time. It uses the flood-ahead technique wherein each cell does not look only for its neighboring cell but also the corresponding cells of its neighbors to get the most optimum path.

ALGORITHM OF ITERATIVE FLOOD-AHEAD

- Set micro mouse at the starting of the maze, say (X, Y).
- Check if cell (X-1, Y) is blocked. If yes, then ignore this path, go to step 5.
- If no, check value of neighboring cell and go to step1.
- If no, then go to step 14.
- Check if cell (X, Y-1) is blocked. If yes, then ignore this path, go to step8.
- If no, check value of neighboring cell and go to step1.
- If no, then go to step 14.
- Check if cell (X, Y+1) is blocked. If yes, then ignore this path, go to step11.
- If no, check value of neighboring cell and go to step1.

- If no, then go to step 14.
- Check if cell (X+1, Y) is blocked.
- If no, check value of neighboring cell and go to step1.
- Compare the values in the unblocked neighboring cells of (X, Y).
- Move micro mouse to the cell which has the minimum value and is closest to the center i.e. (0, 0).
- Repeat these steps until micro mouse reaches at the center of the maze.

PERFORMANCE EVALUATION

Performance evaluation is the key aspect of undertaking any research work. So I have evaluated the work and finally concluded by elaborating the efficiency of the work. But before discussing the execution, I first give an overview of the work done and then its relevant efficiency. The existing framework makes the Micro Mouse move towards center from the starting point. The maze is numbered in an increasing order as we move from center to the outside of the maze. The Micro Mouse then observes the neighborhood cells and the number assigned to them to move forward in the maze. However, it does not intelligently scan the neighboring cells to move forward and hence produces ambiguity in obtaining the most optimal path.

Optimizing the given problem statement means applying and integrating the exiting techniques with other techniques to produce more optimal results. An optimized solution has been presented in this section. The following snapshots are from the earlier execution of Micro Mouse using flood-fill algorithm and flood-ahead algorithm in comparison with iterative flood-ahead algorithm combined with directional algorithm:

Figure 1(a): Execution of Flood-Fill

Figure 1(b): Simulation of Flood-Fill

Figure 2(a): Execution of Flood Ahead

Figure 2(b): Simulation of Flood Ahead

In the above snapshots we can clearly see the ambiguities arising in both the algorithms wherein the Micro Mouse has to travel extra distance to reach the center. On applying and combining iterative flood-ahead algorithm and directional algorithm, we not only we helped in saving the distance travelled by Micro Mouse but also saved the time to reach the center (destination). The following screen shots show the execution of iterative flood-ahead algorithm and directional algorithm:

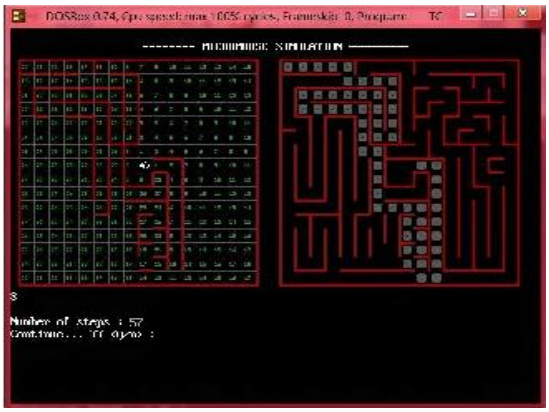

Figure 3(a): Execution of Iterative Flood-Ahead **Figure 3(b): Simulation of Iterative Flood-Ahead**

Clearly, combination of Iterative Flood-Ahead algorithm and Directional Algorithm results in lesser number of steps Micro Mouse has to travel as compared to Flood-Ahead algorithm and Flood-Fill algorithm. Also, as the distance is directly proportional to time, Micro Mouse will take less time to reach the destination, thereby reducing time complexity as well. The working of flood-ahead algorithm in the above snapshot, in place of ambiguity is explained as follows:

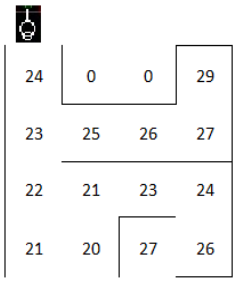

Figure 4: Ambiguity Resolution through Iterative Flood-Ahead

In the above figure as the Micro Mouse travel from cell 23 to 25, i.e. towards the center it follows the wrong path as there is a dead end ahead of it. In the proposed iterative flood-ahead algorithm and directional algorithm the Micro Mouse will follow the natural steps of Flood-Fill algorithm but will travel lesser number of steps. The Micro Mouse halts at cell 23 and uses look-ahead algorithm iteratively till cell 26 for a clear path ahead. On receiving false (Boolean value), as there is a wall (sensed by IR sensors) head of cell 29, it will not traverse the additional four steps it would have traversed earlier (26-27-29-27-26). On moving forward, it will use directional algorithm at cell 22 to choose from same neighboring cell values (i.e. 22) and make the best possible choice for reaching center of the maze. Hence, saving the time and minimizing the distance the Micro Mouse will have to travel. Flood-Ahead algorithm optimizes the distance, by eliminating the miscalculated steps; the Micro Mouse has to travel to reach its destination.

RESULTS

The proposed technique is tested and evaluated by using a sample maze and the outputs are shown in Figure (3). The results obtained from the above experiment have yielded desired results. A tabular comparison on different parameters has been done between different tracing techniques as below:

Table 1

Features	Line-Follower	Flood-Fill	Flood-Ahead	Iterative Flood-Ahead
Efficiency	54.1	97.5	98.9	99.2
Time to Solve(min)	14.5	8.25	7.5	6.75
Variation In Degrees Turned (%)	1.2	1	0.95	0.8
Avg. no. of steps before collision	Infinite	150	120	75
Avg. time per cell (min)	Infinite	N/A*	2.13	1.70

*Varies from maze to maze

CONCLUSIONS

This paper presents a new idea for optimizing Flood-Fill algorithm used in Micro Mouse puzzles. From the achieved results it can be concluded that the combination of Iterative Look-ahead algorithm and directional algorithm we have achieved an efficient way to reach the center of the maze and is useful for reducing the number of steps travelled during traversing. In this work, Look-Ahead algorithm has been applied iteratively in association with directional algorithm to check the cells surrounding a particular cell. A merging between the Look-ahead algorithm and directional algorithm is done and gives very acceptable results. The proposed technique gives a high flexibility to users to travel the most optimum path among several alternatives as shown in Figure (3). With the proposed work, the effectiveness of optimization has been studied carefully. Further investigation to the topic reveals that iterative flood-ahead algorithm can give good results. The concept has been worked out and can be used in future with flood fill algorithm.

REFERENCES

1. Lee, C.Y., "An Algorithm for Path Connection and its Applications", *IRE Transactions on Electronics Computers, 1961.*
2. Ahmed N. Mosa Ayman M. Saad Moustafa A. Mohamed, "MicroMouse: An Intelligent Maze Solver Robot" *IEEE Egypt section, September 2005.*
3. Akers, S.B., "A Modification of Lee's Path Connection Algorithm", *IEEE Transactions of Electronics Computers, February 1967.*
4. http://www.micromouseinfo.com/introduction/mfloodfill.html[accessed on August 2013]
5. Sandeep Yadav, Kamal Kumar Verma, Swetamadhab Mahanta, "The Maze Problem Solved by Micro mouse", *International Journal of Engineering and Advanced Technology (IJEAT)*

   ISSN: 2249 – 8958, Volume-1, Issue-4, April 2012.
6. Gorden Mc Comb, Myke Predko,"Robot Builder's Bonanza", Mc-Graw Hill, 2006 [accessed on August 2013]
7. Work in Progress – Undergraduate Interdisciplinary Research – *IEEE Micromouse Competition*, Robert D. Milos, Allan J. Hall, Neal Shine, Kyle D. See, Tyler R. Bednarz, Yonglian Wang; Department of Electrical and Computer Engineering and Computer Science Ohio Northern University [accessed on August 2013]
8. http://www.google.com/url?q=http%3A%2F%2Fmadan.wordpress.com%2F2006%2F07%2F24%2Fmicromouse-maze-solving-algorithm%2F&sa=D&sntz=1&usg=AFQjCNF0x7vp7y7STl2vZ-PVHlyu4f6fcQ
9. http://en.wikipedia.org/wiki/Micromouse [accessed on September 2013]
10. Garima, Vipul Aggarwal; *Engineering and Technology: An International Journal*
11. http://homepages.uel.ac.uk/C.Kanesalingam/miceintr.htm [accessed on September 2013]