Implementation of the Trémaux Maze Solving Algorithm to an Omnidirectional Mobile Robot

Conference Paper · January 2014

Kai Li Lim
University of Western Australia

Implementation of the Trémaux Maze Solving Algorithm to an Omnidirectional Mobile Robot

Lim Kai Li
Sunway University
Petaling Jaya, Selangor, Malaysia
12057642@imail.sunway.edu.my

Fawaz Y. Annaz
The University of Nottingham Malaysia Campus
Semayih, Selangor, Malaysia
fawaz.annaz@nottingham.edu.my

Abstract

Real time merging of a virtual maze to a mobile robot allows for rapid navigation algorithm development for any foreseeable/target application with minimal cost and time. The Trémaux maze-solving algorithm was implemented. Since this paper aims for revealing the shortest path for navigation, the robot is capable of eliminating double treaded paths, hence increasing the efficiency of maze solving for situations such as search and rescue missions.

Keywords: Mobile robot, maze-solving, virtual maze

1. Introduction

The Trémaux maze-solving algorithm requires the robot to record its path throughout its navigation routine for finding its path. For each path in the maze, the path can be unmarked, marked once, or marked twice. Each of the path marked twice will be bidirectional – once forward and once the other way around, and often lead to dead ends, these paths will not be threaded by the robot for a third time and will be treated as a dead end. The robot will prioritize paths without markings before threading on paths marked once to find another unmarked path [1]. If a solution is found, the path that is marked only once indicates a path that leads straight to the entrance, thus finding only one solution. If the solution is not found, all the paths in the maze will be marked twice. In this paper, the Trémaux algorithm will be implemented on a virtual-maze environment, which is then fed to the omnidirectional robot. Literatures such as [2] has utilized the Trémaux algorithm for verification in exploring dynamic mazes.

2. The Virtual Maze Environment

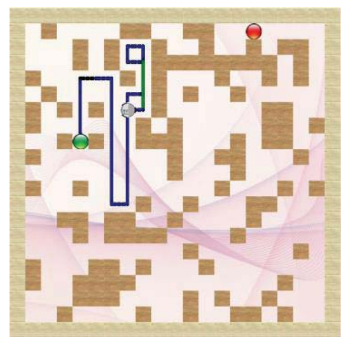

Figure 1: A virtual maze environment

Figure 1 shows a maze that was developed using a Visual Basic program and when executed it allows the user to generate a maze with an initial robot position and final goal locations. The maze can be randomly or manually generated by placing blocks. Communications between the robot and the interface allows the unveiling of the maze to the robot and the tracing of the robots location and movement on the maze.

The robot's Start location is marked with a green dot and a solid line indicates its path and the goal is marked with a red dot. The robot current position is marked with a circle with an arrow that depicts the robot current facing direction. To navigate the maze, two main processes take place, the first is computer based, and the other is mobile robot based. The computer discretely reveals the surrounding locations to the robot, the robot decides on the next move (using Trémaux's algorithm) and reports its new location to the computer, and the cycle repeats, until the goal is reached.

3. The Omnidirectional Mobile Robot

In this paper, the robot makes decisions (that are based on local information revealed to it) and navigates its way through the maze. Manufactured by Festo Didactic, the Robotino [3] is the mobile robot used for this paper. It is built based on the omnidirectional drive assembly for its maneuvering purposes. The robot offers an array of connectivity options including USB, Ethernet and Wi-Fi. The robot maps its movement to onboard memory and continues to navigate and discover its goal using virtual environmental data and marks its path while performing the navigational process. The virtual maze is revealed to the mobile robot on a grid-by-grid basis as the robot moves to a new grid. The Trémaux algorithm developed for path mapping is also capable of finding a sure-solve solution to any maze. This omits dead end paths and redundant loops. To maintain this two-way-update links with the control station (PC in this case) and a final path whisper with other posse robots, Wi-Fi communication is used.

4. Results and Comparisons

The wall-following algorithm is the simplest algorithm employable for solving mazes. It involves a person following a wall, either the left or right wall, until he reaches the maze exit. The logic behind this algorithm is such that the person will always turn right or left, depending on which side the algorithm is used, at all intersections until he meets a dead end, in which case he will turn around and continue to employ the algorithm. The biggest flaw of this algorithm is that it is unable to solve any open-ended maze, meaning that all of the maze that it solves must contain straight, closed path. If a path is started at a wall island, i.e. the path loops around the wall, then the person will run around that wall in circles and not leaving it. (See Figure 2)

(a) (b)

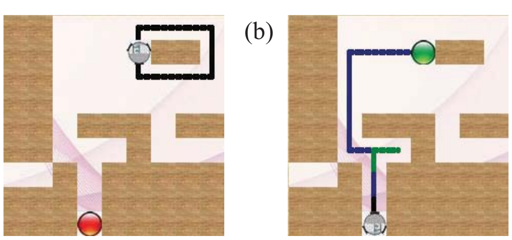

Figure 2: (a) a robot circling an island on a right wall-following algorithm and (b) the same maze solved with Trémaux's algorithm

The Trémaux algorithm shows significant improvements over the shortcomings of the wall following algorithms. This time, the colored marked paths of the algorithm is evident and shows the path taken by the robot, as shown in the figures below.

(a) (b)

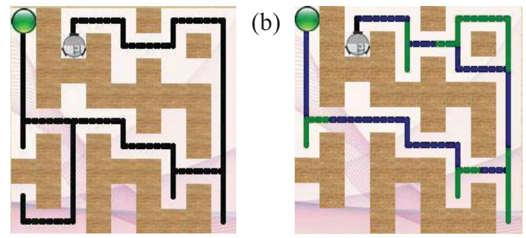

Figure 3: Maze solved with (a) right-wall following and (b) Trémaux's algorithm

The shortcomings of the wall following algorithms where the robot loops an island can be solved using Trémaux's algorithm.

5. Conclusion

In the presented work, two maze solving algorithms were used to solve mazes generated by the user; the wall-following algorithm and the sure-solve Trémaux algorithm. A solution was given to preserve the advantages of proximity navigation whilst avoiding being in an unsolvable loop. Path elimination was also presented with the aim that it will be transmitted to guide a further support robot to the goal.

This paper mimicked a robot discovering a victim using hybrid navigation modes and calculated a sure-solve path that is to be conveyed to personnel wirelessly. This enables further efforts to be done following the marked to prevent meaninglessly backtracking on its own redundant initial discovery path such as in search and rescue operations.

References

[1] M. Babula, "Simulated maze solving algorithms through unknown mazes," in *XVIIIth Concurrency, Specification and Programming Workshop*, Krak´ow-Przegorzały, Poland, 2009, pp. 13-22.

[2] A. Kumaravel and K. Rangarajan, "Constructing an automaton for exploring dynamic labyrinths," in *Radar, Communication and Computing (ICRCC), 2012 International Conference on*, 2012, pp. 161-165.

[3] Festo. (2013, October 18). *Robotino*. Available: http://www.festo-didactic.com/int-en/learning-systems/education-and-research-robots-robotino/