MazeRunner

Submitted By:

Priyank Palod (13CS30046)

Barnopriyo Barua (13CS30009)

S. Raaga Priya (13CS10043)

Abhishek Saini (13CS10001)

Abhishu (13CS10002)

Abstract

In this project, we create a tool called MazeRunner. A tool to correctly find a way through a line maze using various Image Processing techniques. This can serve as a very important tool in many Artificial Intelligence problems, mainly those involving autonomous vehicles for many tasks such as coverage, exploration, patrolling etc. Another important application of this tool can be thought of during creating mazes for checking their solving complexities.

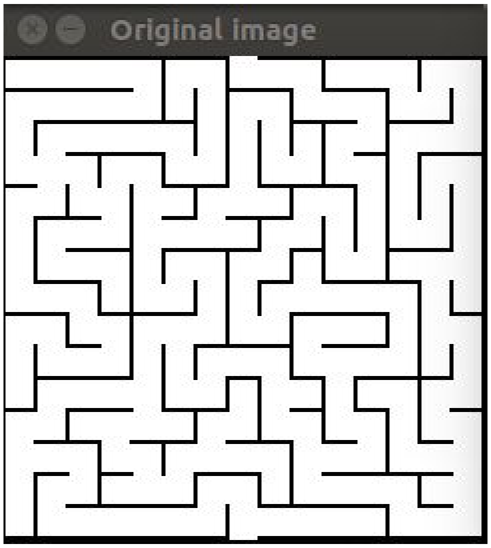
Problem Definition

The Problem can be solved by finding the biggest connected component in the image and then moving along it, i.e. following the wall. This works since the solution path always lies along at least one connected component and not inside it. Since if it was inside the connected component, then that would mean that it has to be closed. Now, usually the path in a maze will always be along the biggest connected component and not the smaller ones, as the smaller components are simple small lines embedded in the maze to confuse the viewer. So the problem simply reduces to marking the path along the biggest connected component.

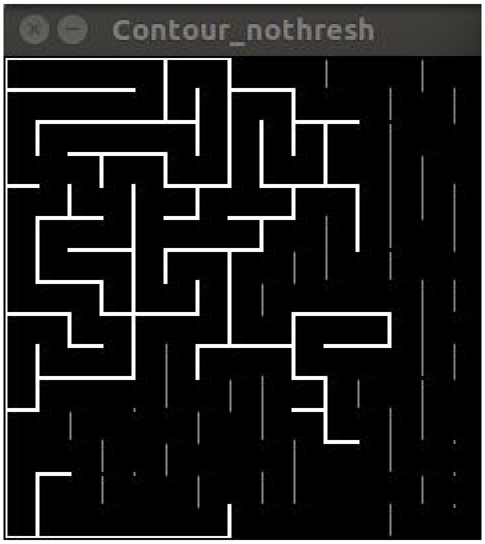
Algorithm

The algorithm is a modified and more general form of wall follower algorithm [1]. The original wall follower algorithm works only for simply connected mazes. In case, all the walls of the maze are not connected together, the algorithm falls apart. So, we thought of a simple but efficient solution that takes in consideration non-simply connected mazes too. Instead of taking any one contour as our main component, we selected the biggest contour as our main component. This makes sure that a small non-connected wall is not selected as the main contour. The algorithm that we devised is described below :-

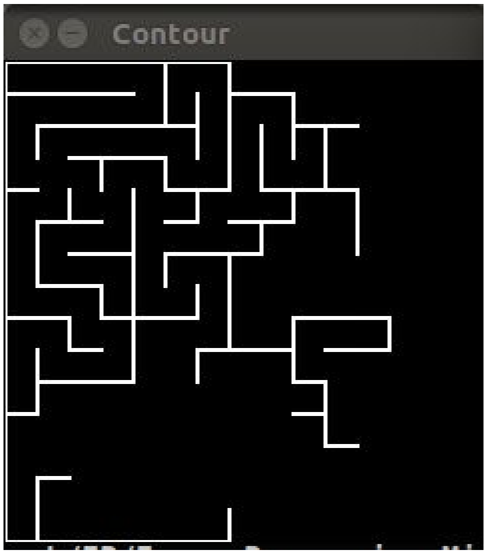
1. Take the image as input and convert it to binary along with inversion.



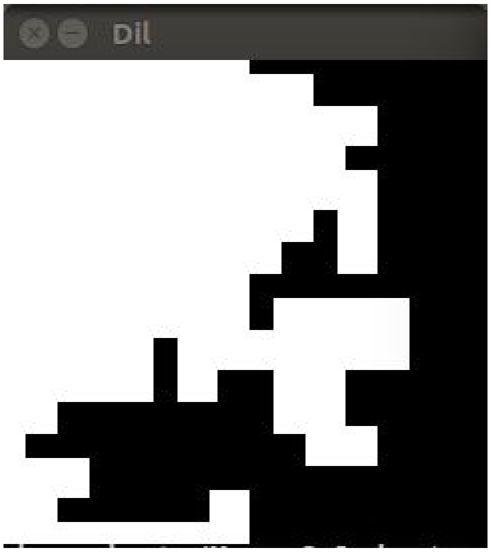
2. Find contours in the image.



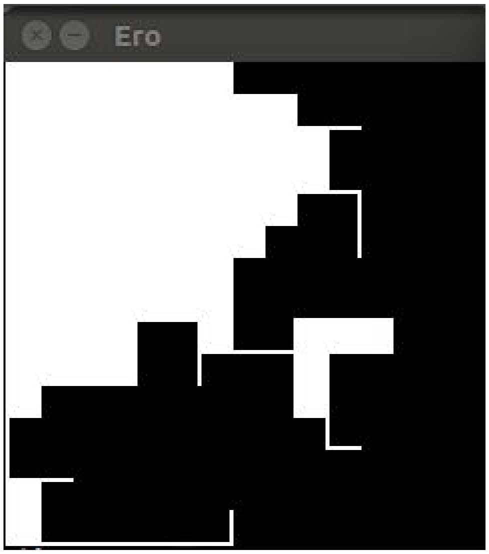
3. Take the largest contour and take just that in the foreground.



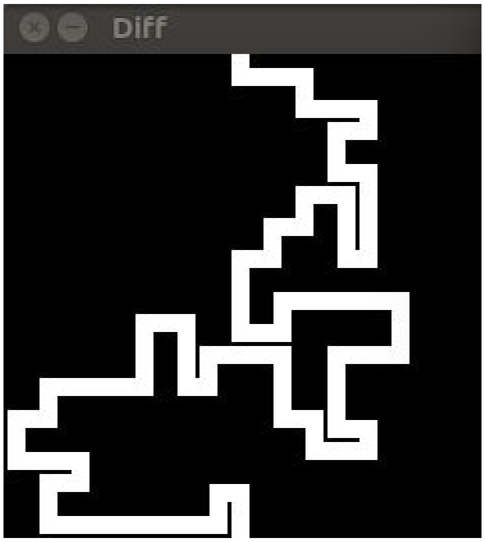
4. Dilate the image using a sufficiently large kernel.



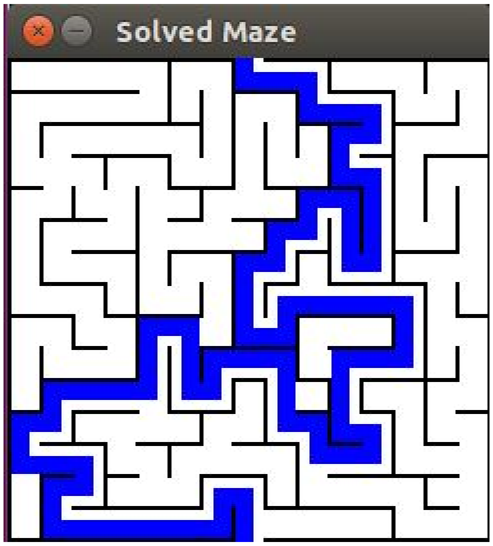
5. Erode the image using the same kernel.



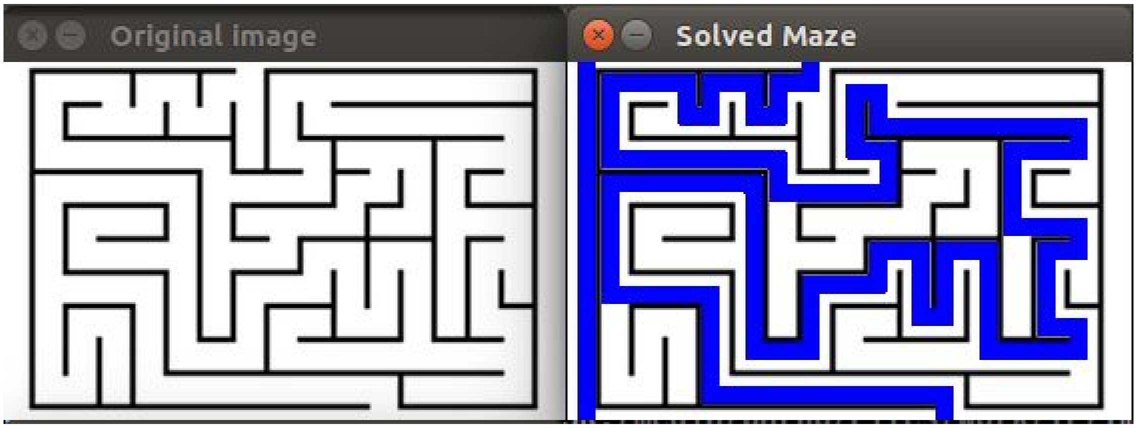
6. Subtract the eroded image from the dilated one. This gives us the path along the largest contour.

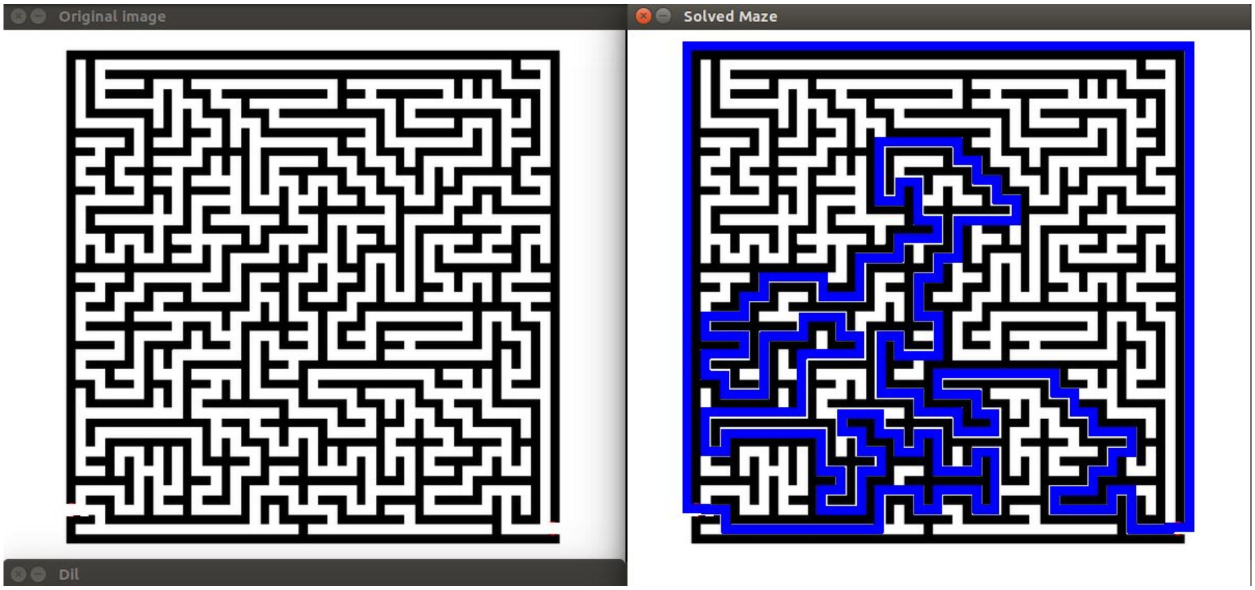


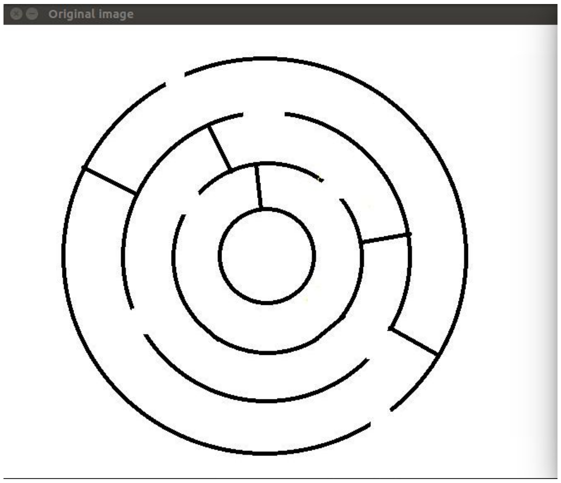
7. Use this image’s inversion as mask to remove only the red and the green components of the image. This colors the path in blue color. Show the resultant color.

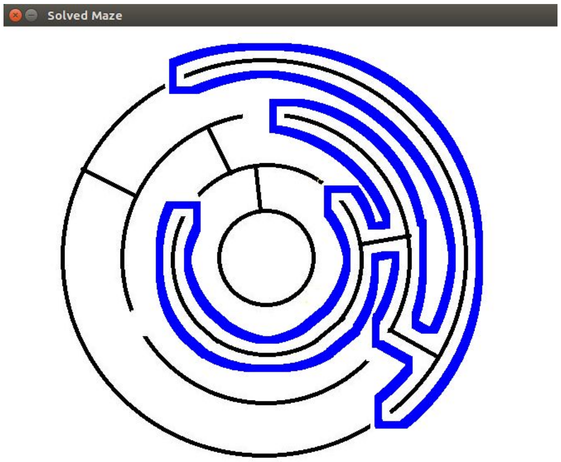


More Screenshots









References

[1] https://en.wikipedia.org/wiki/Maze_solving_algorithm#Wall_follower