A Real-Time Maze Solver for Android

Rohan Paranjpe rap2363@gmail.com
Armon Saied armons@stanford.edu

We would like to develop a real-time maze-solving app for the Android mobile platform. The app would be able to analyze an image of a general maze and use image processing algorithms to binarize the image into a black-white representation, where black pixels are the walls of the maze or non-traversable areas, and the white spaces are part of the path. The app will be able to extract a full-path graph of the maze and solve it (using recursive path-searches or dynamic programming), create a skeletal structure of the solution path, and then overlay the solution back onto the original image.

The real difficulty of the algorithm will be in the detection of the maze vs. path space and being able to binarize the image in real-time using time-averaging methods. Another processing difficulty will be in overlaying the solution in real-time; one of three modes can be specified by the user, a partial solution, the full solution, or a full solution that is revealed gradually (to show how the solution is being drawn out). A rough solution sketch of some of the image processing that would take place would be as follows:

1. Time-average the frames that are inputted to the phone into an image.

2. Binarize image by identifying walls of the maze (generally by assuming the darker regions to be walls and lighter regions to be free spaces), and blackout all areas that are outside the walls (so that we only work inside the maze).

3. Use erosion and minimum area techniques with region identification to do away with small, isolated white spaces.

4. Create a skeletal path structure using morphological thinning which shows the entire path (including all dead-ends) of the maze laid out [1].

5. Identify the start and end of the maze.

6. Convert this skeletal path to a graph representation and use dynamic programming or pruning to find the optimal path from start to end and have a "solution" path.

7. Overlay this solution path onto the phone screen in real-time.

In our preliminary glance at past literature we have found that not much research has gone into the specific problem of real-time conversion of a maze image into a binary representation for computer manipulation. However, we did see several useful studies on using image processing on mazes. One group used a host of image processing techniques for a robotic labyrinth solver [2]. This study spent some time on dealing with the problem of image perspective, but for our project we expect to focus solely on top-down image perspectives, in the interest of time. Furthermore, in choosing maze image pre-processing algorithms we will look to the study on automated maze solving for inspiration. This particular paper details a method of image segmentation based on thresholding an HSV image [3]. With regard to the maze solving aspect of this problem, we may consider a shortest path algorithm, such as Dijkstra's, discussed in this study on solving maze images [4].

References

[1] R. Fisher, S. Perkins, A. Walker, and E. Wolfart. Lecture Notes on Morphological Thinning, 2003.

[2] B. Rahnama, E. Atilla, and M. Shadi. An image processing approach to solve labyrinth discovery robotics problem. In *IEEE 36th International Conference on Computer Software and Applications Workshops*, 2012.

[3] P. Venkata, S.K. Bose, M. Dinesh, P. Sarode, and K. Shaik. Automated Maze Solving using Fluid Mechanics based numerical approach. In *International Conference on Image Information Processing (ICIIP 2011)*, 2011.

[4] M. Yoshitaka and M. Yoshihiro. A Study of Shortest Path Algorithms in Maze Images. In *SICE Annual Conference*, 2011.