EGN 4060C Lab 3: Path Planning

Dr. Gita Sukthankar

Due Oct 1, 2014

In this lab, you will implement the wavefront planner that we discussed in class.

- Write a path planning function using the wavefront algorithm to navigate from a start state to the goal state of a maze. The output of this function should be the path cost map for every square of the maze which you will print out to the screen and use to navigate with.
- Write a function that can navigate from start to goal using a cost map. When you reach the goal, spin the robot in a circle.
- Use the new MapGUI class to display the map and show the robot's movement on the GUI.

Your program will be evaluated on the following things:

- Correctly generating the final cost map for the maze. (1 pt) Please include a printout from your program showing the cost map in your lab writeup.
- Having the robot follow the path from start to goal. (1 pt)
- Showing the robot's movement on the GUI (1 pt)
- Generating a cost map and path for a different starting and goal position in the same maze (2 pts)

You can assume either 4-square or 8-square connectivity. If you assume 8 square connectivity the robot should move accordingly (using the diagonals to move.

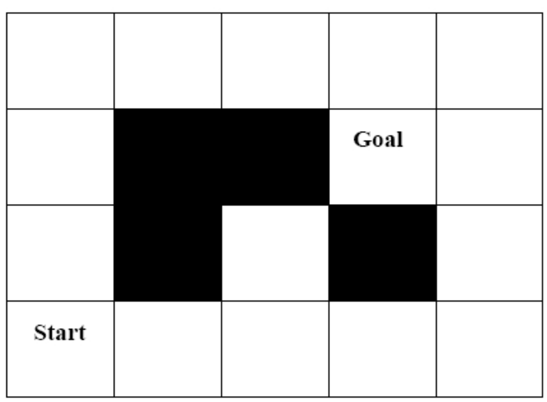

Figure 1: Maze map

Here is an outline of one way to implement a wavefront planner.

- Create two arrays: old cost map and new cost map.
- Initialize the old cost map's goal cell to 2, obstacle cells to 1, and all other cells to 0.
- Iterate until the robot's start cell in the new cost map has a non-zero value (or until the map stops changing).
- For each cell in the old map:
  - If the cell contains an obstacle (1), copy it over to the new cost map.
  - If the cell contains a goal (2), copy it over to the new cost map.
  - Find the lowest cost neighboring cell with a value of 2 or greater.
  - If no such neighbor exists, copy the current value to the new cost map.
  - Otherwise, add 1 to the lowest cost neighbor and copy it to the new map.
- At the end of the iteration, copy the new cost map to the new cost map.

To follow the path, follow the cost gradient—-always have the robot always move to the lowest-cost non obstacle cell.