CS26410 Report 3
Hide and Seek

Chris Savill
chs17@aber.ac.uk

April 13, 2013

Contents

1 Description of algorithms used

1.1 Depth-First-Search (DFS) algorithm

The DFS algorithm is used to aid in exploring a whole area. The algorithm requires that a stack is used to store the path along which the robot has travelled to get to its current cell on the occupancy grid. The reason it acts as a DFS is because the algorithm keeps the robot searching further rather than restricting it to searching a close area to a start point and then working out.

The algorithm has a way of determining if the mapping of an area is complete. The check is a function that evaluates if every cell that has been explored and has no obstacle present has no unexplored neighbours. If this return true then the search and mapping is complete so the path stack is emptied thus finishing the mapping task.

The way in which the algorithm chooses which direction to continue in at each cell is quite simple. If the cell directly in front of the robot is explorable, the robot will continue in that direction. This was implemented as it speeds up the search because the robot takes a long time to turn. If the cell in front has an obstacle present, a random number is generated that correlates to each of the 4 directions it can travel and checks if that cell is unexplored and has no obstacle present. This random number generator is constantly generated until a cell that is unexplored and is explorable is chosen.

1.2 Localisation algorithm

The localisation algorithm is use to determine whether the robot is within the occupancy grid and if so, where. The algorithm itself works as follows. A new temporary occupancy grid is created for the robot to use for its mapping of its immediate area. Once the immediate area is scanned the temporary grid is compared to all areas of the original occupancy grid. The comparison algorithm uses a 2 for loops to ensure that the temporary grid area is compared to all possible areas of the original grid. The temporary grid is compared to areas of the same size and each individual cell is compared with the corresponding cell of the area being compared with in the original grid.

If the localisation of that area is unsuccessful then the algorithm exits telling the user that the robot is not within the original occupancy grid area. If there is more than 1 place the robot could possible be, the robot uses the DFS algorithm and moves to the next unexplored cell and tries to localise with the additional information gathered. If the comparison algorithm determines that there is only one possible place for the robot to be, the robot switches to using the original occupancy grid and is localised to the area it was found in. As localisation requires that an original occupancy grid exists to compare with, there is a boolean value that tell the robot that it mapped successfully so localisation can be attempted.

1.3 Hiding algorithm

The hiding algorithm consists of 3 parts; the first part involves finding the best place to hide, the second part involves finding the shortest path to that position and the third part involves getting the robot to travel along that path to its hiding location. The hiding function can only be accessed if the localisation of the robot is successful.

In order to find the best hiding spot for the robot the algorithm first searches for empty, explored cells and checks if the have 3 walls around them, if none can be found does the same but looks for 2 walls and then 1. The path finding algorithm is a simple A* search that keeps track of the distance/cost so far and the distance/cost left. The search tries to find the shortest path possible as the search is guided in the direction of the target cell and the total goal value is underestimated not overestimated so it is admissible. Once the path as been plotted from the robot's current location to its hiding location, the robot moves cell by cell along its path until the path stack is empty, at which point the robot should be at its hiding location.

1.4 Seeking algorithm

The seeking algorithm is very simple, if the robot localised successfully, the robot will perform a DFS of the whole pre-explored area, if however any cells that did not contain an obstacle on the original occupancy grid are detected to have an obstacle, then the DFS finishes and the anomaly is stated to the user, thus finishing the seeking function.

2 Trial runs

2.1 Simulated trial runs

2.2 Real world trial runs

3 Discussion of trial run results

4 Evaluation of work done

4.1 Positive parts

4.2 Negative parts

4.3 Hardest parts

4.4 What would be done differently