

Kevin Dombrosky

Lab 1 Writeup

Preface – Ran out of time to complete the assignment in its entirety between the Career Fair, Interview Day, other exams, and being flown down to Pittsburgh for an interview the day before the due date. I know it's really no excuse but since this isn't really like me, I felt like I had to say something.

Cost function:

The cost function was implemented to calculate the cost that it takes the person to move from one pixel on the map to the next pixel in the path. It begins by calculating the average time for a person to cross the distance of the pixel transition based on a time per distance value that I found online. This time per distance value assumes the person is unhindered by elevation change or terrain. Then, based on the terrain of the source pixel, a multiplier is assigned to the cost that simulates it being harder to move through a particular terrain. The multiplier will never be less than one. Then another multiplier is assigned based on the elevation change per distance between the two pixels. Again, this multiplier is never less than one. The cost function then returns the actual cost plus the heuristic cost.

Heuristic function:

The heuristic function returns an infinite distance for any pixel that is marked as impassable vegetation, water, or out of bounds. This essentially says the pixels are invalid, and will never be visited by the A* algorithm. It then returns the average time for a person to cross the distance of the source to target in a straight line. Since my cost function will only ever increase the cost of a pixel transition through the multipliers, and the shortest distance between two points is a straight line, the heuristic will never overestimate the cost of a path. Terrain and elevation cannot really play a role in the calculation of the heuristic value since you have no idea on what lies between the two points.

ScoreO:

This is the big piece that I was unable to complete. My approach so far was to essentially perform a breadth first search, once a complete path was discovered or the time limit was reached, I would remove the latest flag's path, add on the path and cost back to the start if the path was not complete, and trace through the parents to build my graphical path. This would yield a minimum result if the path was complete, however not necessarily for if the path was incomplete. Graphical output of the path was not able to be implemented in time.

Human readable output:

The furthest I got on the human readable output was the red path on top of the map image. Once a classic path completed the A* search, go through each pixel that was marked as taken or a part of the path and change it to red before displaying the map.