Project Companion

Gene Harvey

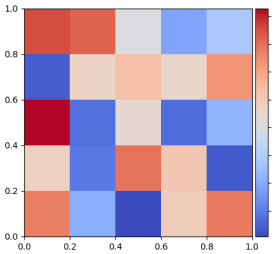
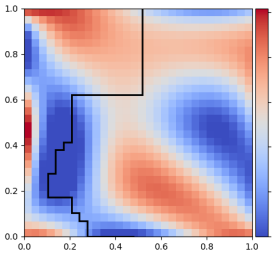
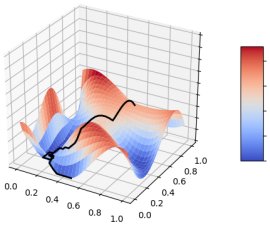
April 22, 2017

Purpose

To parse quadtree data using adjacency lists into a graph visual, and to find the shortest path between all nodes in the quadtree. Selection of closest nodes in the structure enables approximate selection of the shortest path.

Methods

Quadtree relative Euclidean distance data is loaded into a three dimensional hashtable. This forms an undirected graph with weighted edges. We then use the Dijskstra algorithm or the A* algorithm which is the same with a small change to introduce a heuristic. While the Dijsktra algorithm will search almost all possible paths to find the which is most efficient,the heuristic will simply tell the algorithm that points which are farther away from the goal are less likely to produce the shortest possible path. This ensures that less points are ultimately checked, therefore making the algorithm faster.

Optimized paths are found and stored *dynamically* which means that the paths previously found are used to find others. For example if we previously found a path from A to B, and want to find a path from another point C to A, then if at any point the algorithm ends up at B we automatically know the rest of the path (we can do this since the algorithm is continuously finding the shortest path to each successive node from B). This increases speed tremendously.

We also implement an algorithm to quickly find the node closest to an arbitrary point on the grid. This decimates the grid and uses a recursive truncation to determine the grid within which the point lies. The allows us to skip the process of calculating the distance to each node and picking the shortest [I may put more here later].

Finally, the process of finding the shortest path between all nodes is done in parallel, and this is naturally implemented just by partitioning the set of starting points and ending points.

Challenges

The largest challenge had mostly to do with implementing the quadtree data structure to allow for bijective movement and elimination of nodes within the illegal boxes without causing the entire structure to fail.

We also found difficulty in introducing the dynamic process in an efficient way.