By:
Chris Bruun Mikkelsen
Daniel Haraldson
Steen-Bjørn C.R. Vinther
11/11/2016

ROB5 2016 – Second Mandatory Project

In this assignment the Brushfire Algorithm is used to make a Generalized Voronoi Diagram. This is then used for path planning via Dijkstra's algorithm to make a line between two points and around the obstacles.

Making the Brushfire

The image with the map is loaded. The big oval/rectangular shapes represents the objects that needs to be avoided.

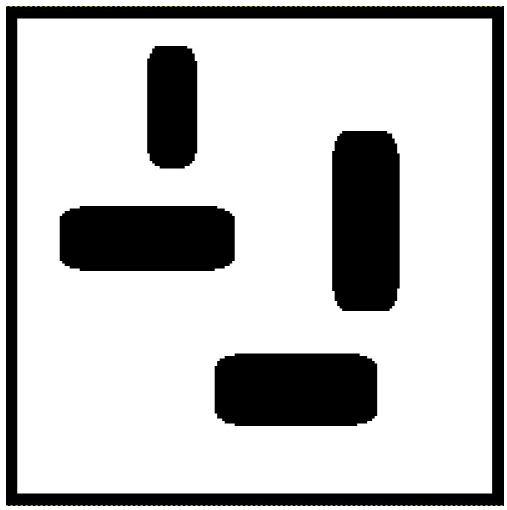

Figure 1: This is the base image that all the other images are generated from.

The edges are then detected for all the object by checking if the point is adjacent to a white spot or not. The points are then colored and the process is repeated. Each time, the outline will grow as the only white spots are away from the object. This is done for all the objects at the same time but each object has it's own outline color. This results in a image where the "waves" of the outline meets.

By:
Chris Bruun Mikkelsen
Daniel Haraldson
Steen-Bjørn C.R. Vinther

Figure 2: Here is the base image with the brushfire diagram.

After that the objects are removed so as to make the process of making the voronoi diagram easier.

Figure 3: Here is the brushfire diagram without the objects.

By:
Chris Bruun Mikkelsen
Daniel Haraldson
Steen-Bjørn C.R. Vinther

Making the voronoi diagram

Then the voronoi diagram is made by looking at the places where the color is different. All the points are the checked. If a point is not the color code(254), it is not surrounded by points that are and one of the points surrounding it is of a different color than itself, it is assigned the color code. Then a new image is made with the previous image but with the color coded points colored black.(This gives an image where the points where the waves collide are black)

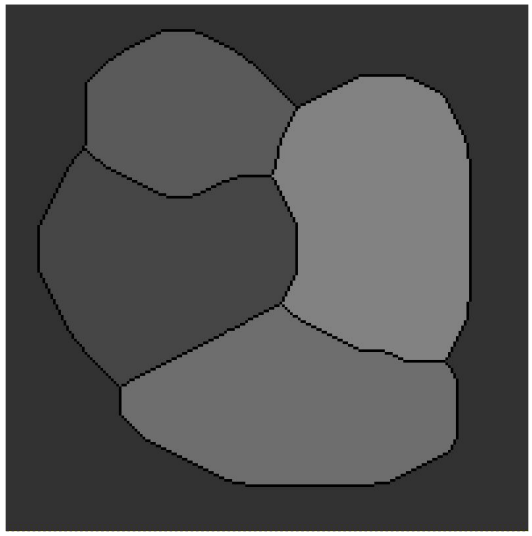

Figure 4: Here are the voronoi diagram.

Figure 5: A image where the distance from the objects is represented by increasing whiteness is also produced.

This is used to connect the start and end point to the voronoi diagram(here seen without the brushfire or voronoi diagram overlay). This insures that no matter where the start point is at the top left and the endpoint is at the bottom right.

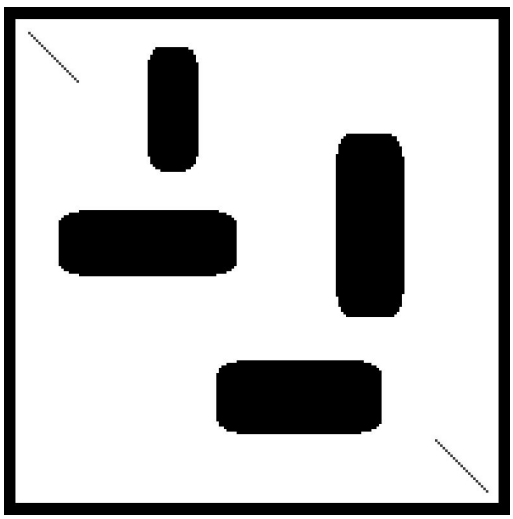

Figure 6: The lines are the path onto the voronoi diagram.

By:
Chris Bruun Mikkelsen
Daniel Haraldson
Steen-Bjørn C.R. Vinther

Finding the shortest path via Dijkstra's algorithm

Then all the points including the connecting lines are used to make a path around the objects from the startpoint to the endpoint.
All the points are put in a vector and the order of this is considered random. From here the dijkstra tree will grow using the starting point as a seed. The tree is made by taking that "undiscovered point" that is closest to the root of the tree, and use it to find new "undiscovered points". This process is constantly repeated until the goal has been reached, or there is no more points to discover. Once case one holds true, the program will find the path by backtracking the route back down to the starting point/root of the tree. This will then generate the path as shown in figure 7.

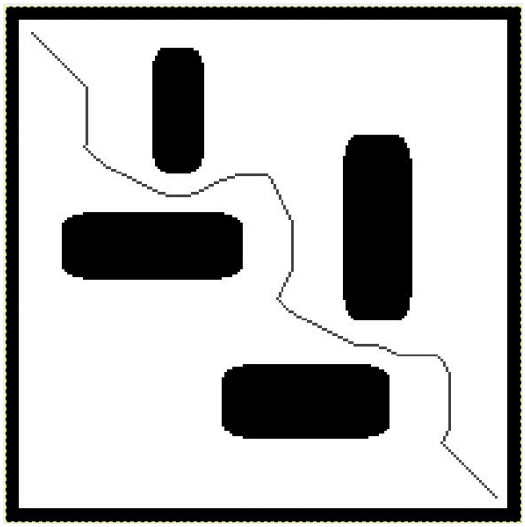

Figure 7: the final path from p1 = (10,10) to p2 = (190,190)