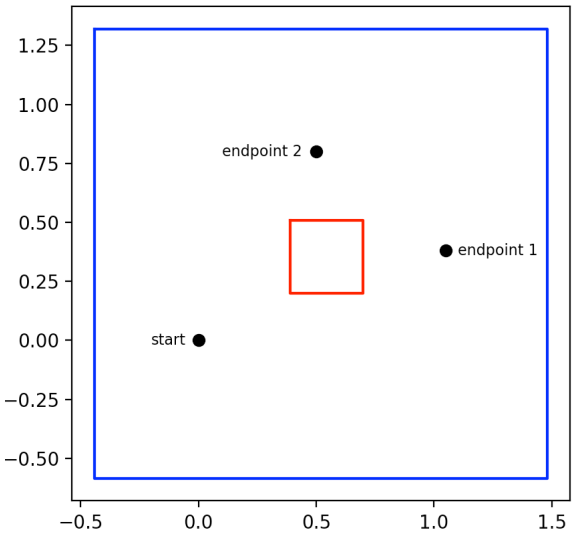
CSE 276 Homework 4
Thomas Liu

Representation

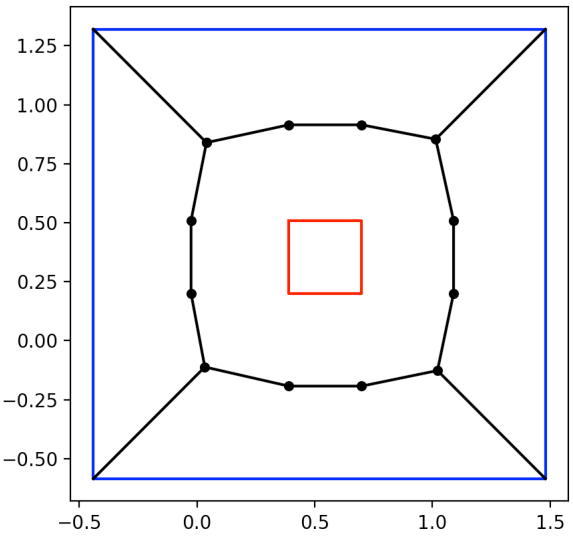
I represented everything in terms of x and y coordinates in meters, with the PiCar's starting position being (0,0) and pointing at 0 degrees. So the obstacle was four points with labels of "topleft", "topright", "botleft", and "botright", indicating the four corners. The boundaries were similarly represented. The overall setup is shown below, with endpoint 1 having coordinates (1.05, 0.38) and endpoint 2 having coordinates (0.5, 0.8).



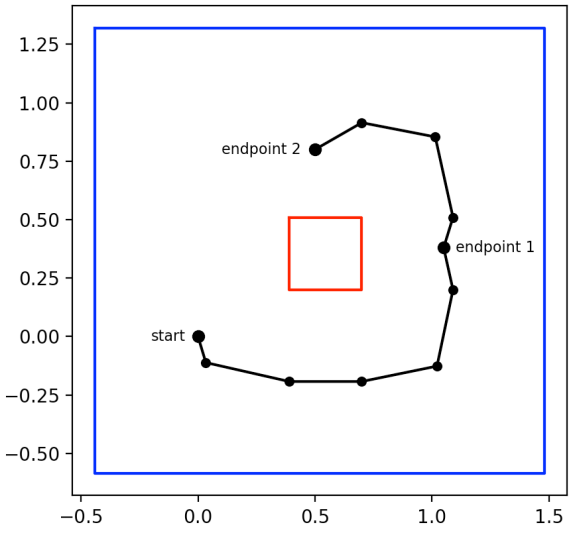
Algorithms

Voronoi

For the Voronoi part, I decided to do a linearized approach, where curves were turned into line segments from one linear part to another. This was partly due to the PiCar movement algorithm which is discussed below. The exact point where the curves met in the corners was equal to the obstacle corner in that area plus or minus an offset. The x-offset was equal to $\sqrt{2 \cdot x_{dist} \cdot y_{dist}} - y_{dist}$ and the y-offset was equal to $\sqrt{2 \cdot x_{dist} \cdot y_{dist}} - x_{dist}$, where x_{dist} is the distance between the obstacle and the boundary in terms of x coordinates, while y_{dist} was the same but for y coordinates. The resulting Voronoi graph is shown below.

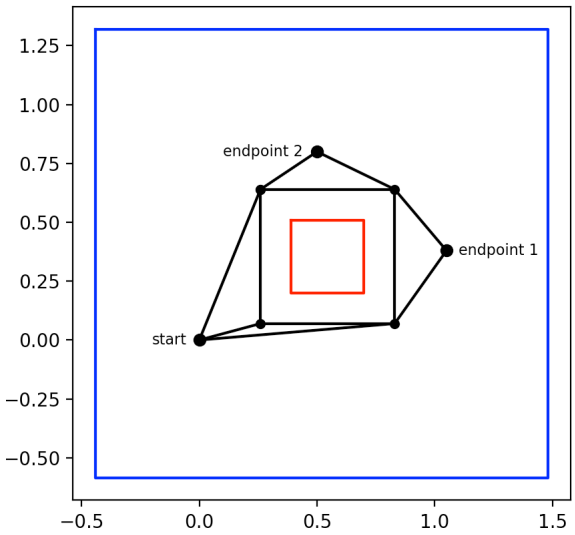


For finding a path from start to end, I first found the two closest intersection points on the Voronoi graph to the start, and also the two closest to the end. I then found the shortest path by applying Dijkstra's algorithm on the four possible combinations of start and end intersection points. The resulting paths (from start to endpoint 1, and endpoint 1 to endpoint 2) are shown below.

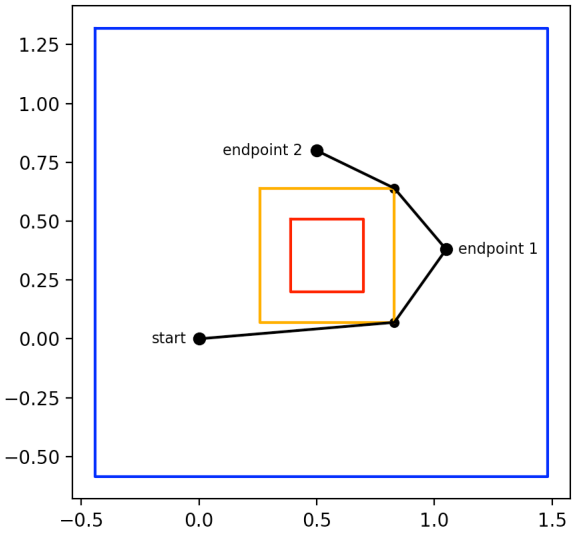


Visibility

For the visibility graph, I first padded the obstacle on all sides by the width of the PiCar. I then created six nodes, which were the four corners of the padded obstacle and the start and end points. I then connected nodes to each other based on whether they would be visible to each other (using the padded obstacle). The resulting graph is shown below.



For finding the shortest path, I ran Dijkstra's algorithm on the graph above. The resulting paths (from start to endpoint 1, and endpoint 1 to endpoint2) are shown below, where the original obstacle is shown in red and the padded obstacle is shown in orange.



PiCar Movement

For PiCar movement, I used a very simple movement scheme. First, the PiCar would rotate in place by repeatedly going forward and backwards with alternating front wheel angles. It would do this until it was pointing towards the next destination. Then, it would slowly move forward until it either got close enough to the destination or passed it.

Difference in Execution

Voronoi

The Voronoi graph based path did not get close to either the boundary or the obstacle, but had to do much more turns, which resulted in it going slightly offtrack by the time it had reached endpoint 1. That's why the execution was split into two parts, with part one showing the PiCar moving from the start to endpoint 1, and part two showing the PiCar moving from endpoint 1 to endpoint 2. In part two, the PiCar starts slightly tilted and not at 90 degrees, since that is the angle at which it is supposed to arrive at endpoint 1 under ideal conditions.

Visibility

On the other hand, the visibility graph based path had very few turns and was able to arrive at both endpoints quite accurately in a single run. However, unlike the Voronoi graph based path, it did end up very close to the obstacle during its execution.

Comparison

Based on these results, the visibility graph method is most suited for when getting to the destination quickly is important. It's also well-suited for when the robot's handling isn't very good. To deal with possible collision, one needs to simply increase the padding amount by a bit.

The Voronoi graph method is most suited for when the exact position of the obstacle is very uncertain. It would be able to avoid the obstacle in most cases by taking the safest possible path, while the visibility graph method would almost certainly crash unless an unreasonably large amount of padding was added.