Robot Motion Planning - Optional Homework

David Akhihiero

October 21, 2024

1 Introduction

In this assignment, I implemented the visibility graph roadmap and sPRM graph (with 200 nodes) on a 2D configuration space with obstacles. I used A* motion planning algorithms to search the graphs from a start node to a goal node and converted the straight-line paths to polynomial trajectories using the method described in Mellinger and Kumar (2011). I assumed that the robot was a holonomic robot on a ground plane capable of independent x and y motion.

2 Graph Creation and Path Planning

I created the visibility graph using my $createVisibilityGraph(new\_obstacles,\ dense)$ function from assignment 2 and the sPRM graph using the functions from assignment 3. The graphs for a dense environment with a start node of $[0.01, 0.01]$ and a goal node of $[0.9, 0.9]$ are shown in Figure 1.

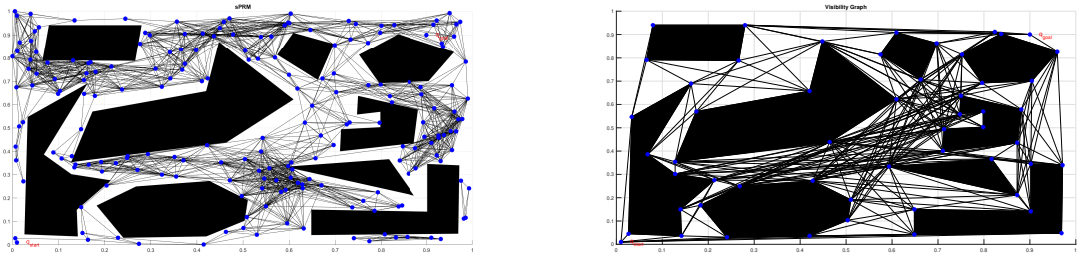

Figure 1: sPRM and visibility graph for a dense environment

I used A* to plan paths in the graphs. The path in the sPRM graph had 12 waypoints including start and goal and that in the visibility graph had 8 waypoints.

3 Trajectory Generation

To convert the straight-line paths to trajectories, I wrote a $getTrajectory()$ function to generate a polynomial trajectory between two waypoints for each coordinate given the order of the derivative of the trajectory being optimized, $k = 4(snap)$, the order

of the polynomial, $n = 5$, the total time, $T = 1$, the position and velocity constraints at both ends of the trajectory and environment obstacles. The position constraints are the coordinates of the waypoints. The starting velocity, u, is 0 and at the end of each waypoint, the final velocity was computed using Newton's law. The acceleration during the trajectory was calculated using Equation 1 then the final velocity with Equation 2

$$s = uT + \frac{1}{2}aT^2$$

$$a = \frac{2(s - uT)}{T^2} \tag{1}$$

$$v^2 = u^2 + 2as$$

$$v = \sqrt{u^2 + 2as} \tag{2}$$

where

- s: length of the straight line path
- u: velocity at the start of the trajectory
- v: final velocity

The final velocity at the last waypoint is 0. The function computes the Hessian, computes the velocity and position equality constraints, and the obstacle inequality constraints. To compute the inequality constraints along a polynomial segment the function steps in increments of $dt = 0.01$ along the straight-line path. For the x trajectory, at each step, it scans the environment at the same y coordinate from the point, p, on the path, to the nearest obstacle on the left and to the nearest obstacle on the right. The inequality constraint is set up so that at that time step, the x of the polynomial is greater than the nearest obstacle point on the left and lower than the nearest obstacle on the right. For the y trajectory, the scanning is from p to the nearest obstacles above p and below p with a similar constraint setup.

The results of the trajectory generation on the paths computed for the sPRM and visibility graphs are shown in Figure 2 and 3 respectively. The path is smoother and completely avoids all obstacles but the curve is sometimes too sharp (in the visibility graph).

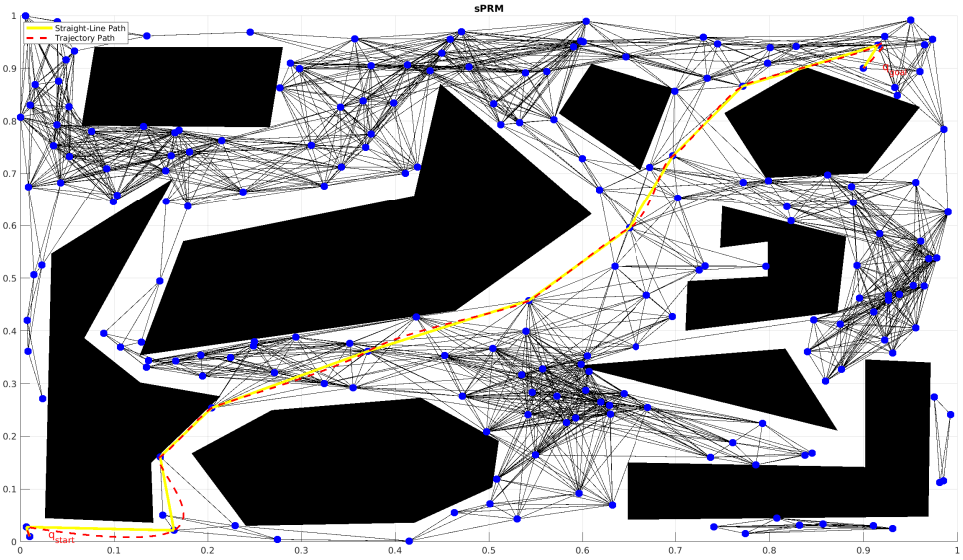

Figure 2: Trajectory vs straight-line path on an sPRM graph for a dense environment

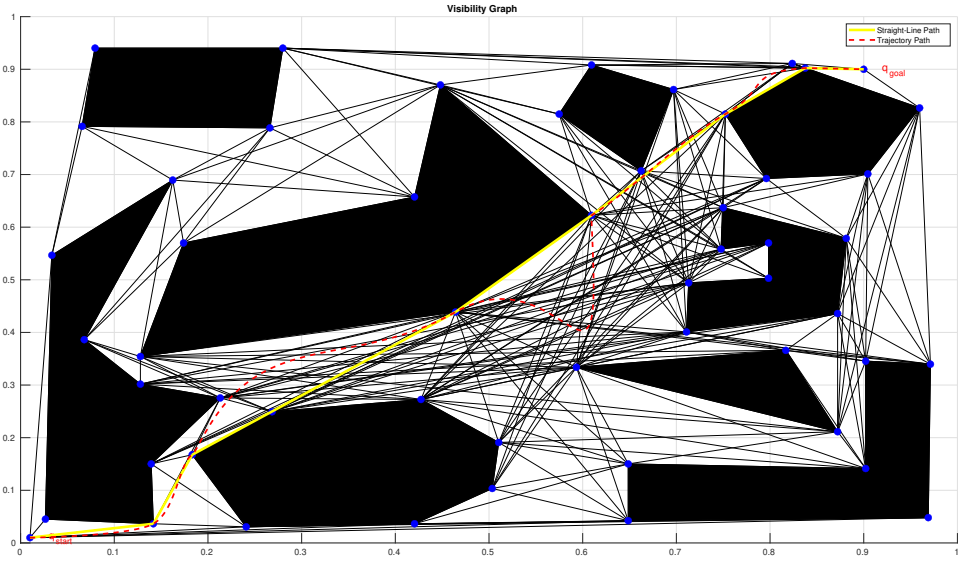

Figure 3: Trajectory vs straight-line path on a visibility graph for a dense environment

References

Mellinger, D. and Kumar, V. (2011). Minimum snap trajectory generation and control for quadrotors. In *2011 IEEE international conference on robotics and automation*, pages 2520–2525. IEEE.