Assignment no. 3

due: Tuesday, December 31st, 2019

All the exercises in this assignment refer to the setting of a line segment (rod, ladder) translating and rotating among (not necessarily convex) polygonal obstacles in the plane.

Exercise 3.1 In class we saw a representation of the free space for the rod-among-polygons problem, the complexity of which is $O(n^5)$. The actual complexity of the free space in this motion-planning instance is much lower.
(a) Show that the maximum combinatorial complexity of the free configuration space in this case is $O(n^2)$. To show this bound you have to bound the number of semi-free triple contacts (namely placements of the rod where it touches the obstacles boundaries in three points without penetrating into the obstacles). You can get a hint (for free) by sending me an email (`danha@post`) with the subject line “hint for Ex 3.1”
(b) Show that the above bound is tight in the worst case. That is, describe a scene where the complexity of the free space is $\Omega(n^2)$.

Exercise 3.2 (p2) Solve the rod-among-polygons motion planning problem with PRM.
(a) Write a brief description of your planner and its major ingredients.
(b) Present experimental results obtained with your program, depicting instances of varying difficulty for your planner, from easy to hard.
(c) Try two different distance measures (of your choice/design) between a pair of configurations. Design, carry out, and report on experiments that show the effect of the choice of distance measure on the performance of the planner.

Exercise 3.3 (optional, bonus, p2) The template program that you got for Exercise 3.2 uses a naive collision detector. Devise and implement a more efficient data structure for collision detection between a fixed length rod and a set of polygons in the plane. The set of polygons is fixed and a query specifies the position and orientation of the rod.

Exercise 3.4 (optional, bonus, you can submit in pairs) Give an efficient algorithm to solve the rod-among-polygons problem, and analyze its complexity.