# Visibility-based Robot Path Planning for a Planar Robot

Joan Gerard, ULB

November 7th, 2019

## 1 Project Description

A robot needs to be capable of building the path from its initial position to a given place in the environment in order to plan its moves and finally get to the goal position.

The problem will be solved using computational geometry techniques but some assumptions will be given in order to limit the complexity of the problem. A static (no people or other robots moving around the environment) 2D rectangle environment will be considered as shown in figure 1. A robot and an obstacle will be defined to be planar with a shape of a polygon not necessarily convex. The robot should be able to plan the shortest path from a given start to an end position coordinate in the environment, avoiding collisions with obstacles and walls. Notice that rotation of the robot is not considered within this problem.

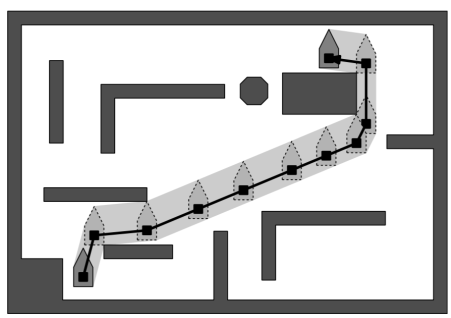

Figure 1: Robot Motion Planning. Image obtained from [1].

## 2 Computability Geometry Techniques Proposal

The problem should be partitioned into sub problems and solved them individually. The technique to solve this problem is based on Lee's technique [2].

All the obstacles in the 2D environment and the robot itself are defined as polygons. It is possible to obtain a new configuration space where the objects are grown in order for the robot to avoid penetrating or colliding within any obstacle while planning the path[3]. A robot $P \subset \mathbb{R}^2$ has a placement reference point at (x,y) denoted as P(x,y). The configuration-space obstacle (known as C-obstacle) of an obstacle $Q \subset \mathbb{R}^2$ and the robot P will be the set of points in the configuration space such that the placement of P intersects Q, denoted as $CQ$, thus: $CQ := \{(x, y) : P(x, y) \cap Q \neq \emptyset\}$[1].

The Mikowski sum between two sets $P \subset \mathbb{R}^2$ and $Q \subset \mathbb{R}^2$ is defined as $P \oplus Q := \{p + q : p \in P, q \in Q\}$. Thus if P and Q are two convex polygons, a brute force algorithm would execute a pair-wise sum between all the elements of P and Q and then create the convex hull of the new set of points running in $O(nm)$ where n and m are the number of elements in P and Q respectively. The algorithm efficiency can be improved to be $O(n+m)$ when the polygons are convex. If we have a convex and a non-convex polygon, the Minkowski sum is bounded by $O(nm)$ since the non-convex polygon can be triangulated, the Minkowski sum can be calculated between each triangle and the convex polygon, and finally the union of all the previously calculated Minkowski sums can be taken to have the final resulting polygon. When both polygons are non-convex triangulate both and apply same technique as above will take $O(n^2m^2)$. This technique is useful to obtain the aforementioned C-obstacle of an obstacle[1].

Compute the shortest path in the new configuration space between two points: $a \in \mathbb{R}^2$ and $b \in \mathbb{R}^2$ that will take the robot from position $a$ to $b$. This is made using a visibility graph[4]. Such graph connects all the vertices of the obstacles that are visible between each other then assign a weight to each edge and using Dijsktra's algorithm the shortest path between $a$ and $b$ is found.

# 3 To Implement

This project will provide an extensive explanation to Minkownski sum constructions (the brute force and the optimized one) for convex-to-convex, convex-to-non-convex and non-convex-to-non-convex polygons and their respective time complexity upper bounds. Many interactive applications are planned to take place during this process demonstrating its usefulness:

1. Brute force algorithm: Let the user create two polygons and construct the Minkownski sum using the brute-force algorithm.

2. Create the C-object of a polygon. Let the user create two polygons P and Q. Show the result of applying $-Q$. Then construct the C-object of $Q$ using $P \oplus -Q(0,0)$. Create an animation in which the polygon Q goes around P showing that the placement reference point corresponds to the bound of the C-object polygon.

3. Illustrate the Minkowski sum construction with the optimized algorithm for convex-to-convex, convex-to-non-convex and non-convex-to-non-convex polygons. Notice that in this part the union of two polygons need to be illustrated together with the triangulation of a non-convex polygon.

Additionally, the project will explain how a visibility graph is constructed.

Finally, the user will be able to create her own environment selecting her robot among a set of pre-defined robots and adding obstacles into it. Then she will select a start position and a goal position and the program will trace the shortest path between a and b moving the robot. The user will have the option to see the C-obstacles and the visibility graph.

# 4 Author's Comments: Motivation

Even though the references of this project are old and the main idea comes from the 80's I found it very interesting for mainly two reasons: first, I have worked on personally projects (none of them using computational geometry) on the area of robot motion positioning before which is related to robot motion planning and I found it exciting; second, since it shows a real application of the many techniques that we saw during class I found it perfect for demonstrating the learnt knowledge on computational geometry.

# References


[1] Mark de Berg et al. *Computational Geometry: Algorithms and Applications*. 3rd ed. Santa Clara, CA, USA: Springer-Verlag TELOS, 2008.

[2] Der-Tsai Lee. “Proximity and Reachability in the Plane.” AAI7913526. PhD thesis. Champaign, IL, USA, 1978.

[3] Joseph O’Rourke. *Computational Geometry in C*. 2nd. New York, NY, USA: Cambridge University Press, 1998. ISBN: 0521640105.

[4] Tomás Lozano-Pérez and Michael A. Wesley. “An Algorithm for Planning Collision-free Paths Among Polyhedral Obstacles”. In: *Commun. ACM* 22.10 (Oct. 1979), pp. 560–570. ISSN: 0001-0782. DOI: 10.1145/359156.359164. URL: http://doi.acm.org/10.1145/359156.359164.