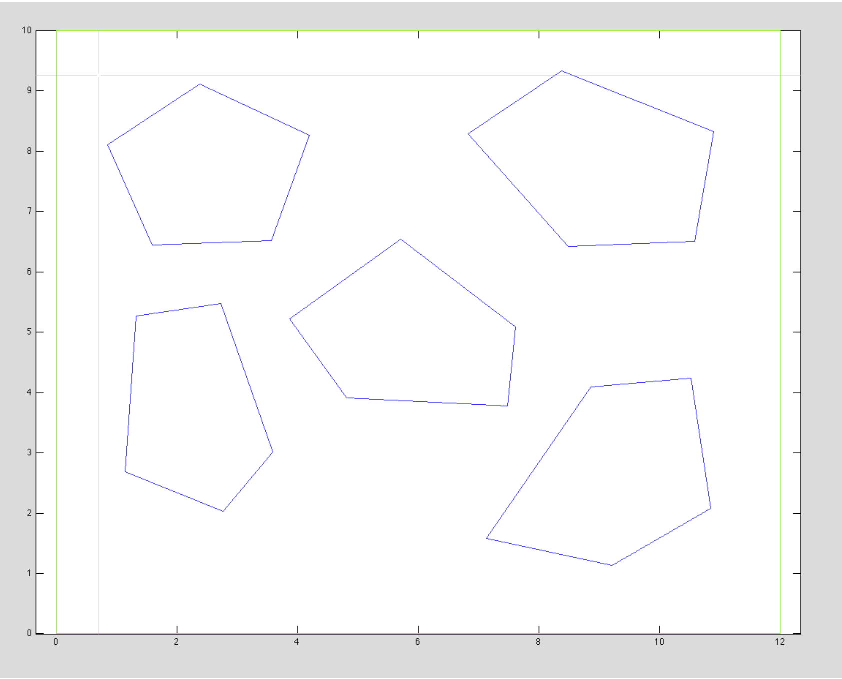
Autonomous Robots

Topological Maps – Visibility Graph

This document will guide you through the practical work related to path planning algorithms based on topological maps. In particular, this practical exercise consists on programming the Rotational Plane Sweep algorithm to build a Visibility Graph. All the code has to be programmed in Matlab.

1. Environment

In order to have a convenient environment, a set of polygons has to be defined. These polygons will have a set of vertices and a set of edges.

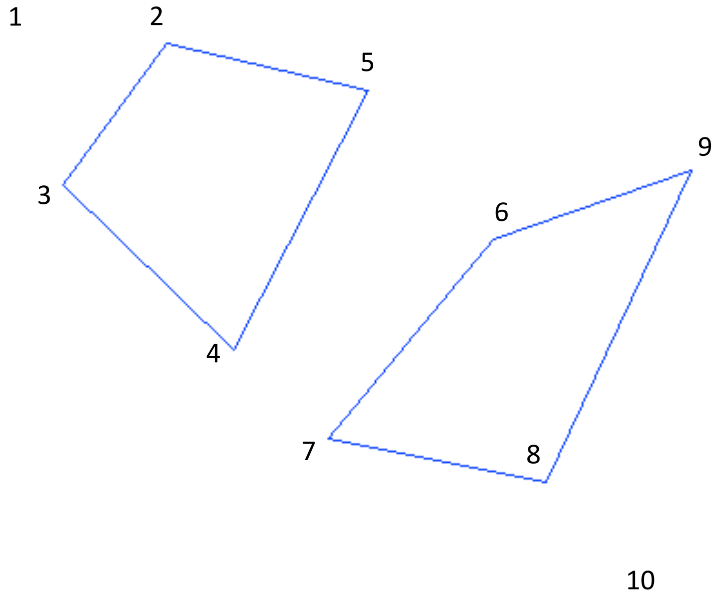


WORK TO DO: Create in Matlab this environment. You can either use some manually stored set of polygons, or create them interactively with the command "ginput". You have to define also the start point and the goal point. Represent the environment with all the polygons and points.

Store the vertices in a Matlab variable called "vertices" using the following format:

- Columns:
  - First column: x coordinate
  - Second column: y coordinate
  - Third column: object number
- Rows:
  - Each row represents a vertex
  - The first row will ALLWAYS represent the START vertex
  - The last row will ALLWAYS represent the GOAL vertex
  - Vertices from the same object will be inserted sequentially. The object can be plotted by joining with a straight segment each vertex with the next one, and closing at the end the object from the last vertex till the first one.

Example of a “vertices” variable corresponding to the following environment:



vertices <10x3 double>

	1	2	3
1	0.7807	9.0497	0
2	3.0322	8.9912	1
3	1.3655	6.7105	1
4	4.1140	4.0497	1
5	6.2778	8.2310	1
6	8.2953	5.8333	2
7	5.6345	2.6170	2
8	9.1433	1.9152	2
9	11.4825	6.9444	2
10	10.2544	0.5702	3

Vertex 1 corresponds to the starting position. Vertex 10 corresponds to the goal position. There are 2 4 vertex objects, vertices 2 to 5 and vertices 6 to 9. The start vertex has been assigned as object 0. The goal vertex has been assigned as object 3.

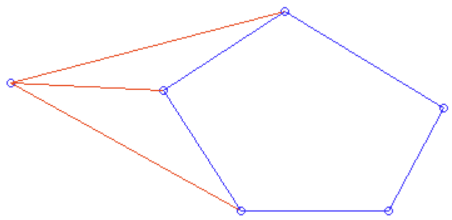
2. Rotational Plane Sweep algorithm.

WORK TO DO: Program the Rotational Plane Sweep algorithm to compute the visibility of the starting point with respect to all the vertices and goal point.

Algorithm 5: Rotational Plane Sweep Algorithm

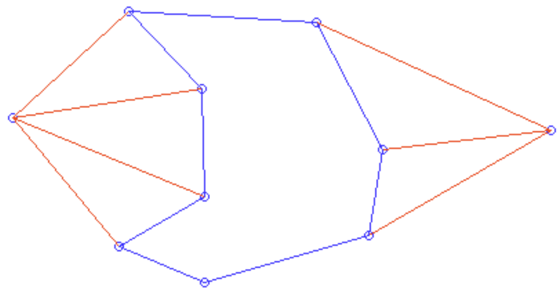
Input: A set of vertices $\{v_i\}$ (whose edges do not intersect) and a vertex v
Output: A subset of vertices from $\{v_i\}$ that are within line of sight of v

1: For each vertex v_i, calculate α_i, the angle from the horizontal axis to the line segment vv_i.
2: Create the vertex list $\mathcal{E}$, containing the α_i 's sorted in increasing order.
3: Create the active list $\mathcal{S}$, containing the sorted list of edges that intersect the horizontal half-line emanating from v.
4: **for all** α_i **do**
5:   **if** v_i is visible to v **then**
6:     Add the edge (v, v_i) to the visibility graph.
7:   **end if**
8:   **if** v_i is the beginning of an edge, E, not in $\mathcal{S}$ **then**
9:     Insert the E into $\mathcal{S}$.
10:   **end if**
11:   **if** v_i is the end of an edge in $\mathcal{S}$ **then**
12:     Delete the edge from $\mathcal{S}$.
13:   **end if**
14: **end for**



3. For a polygon vertex.

WORK TO DO: Test the Rotational Plane Sweep algorithm for a vertex which is part of an obstacle polygon. Check that your implementation works correctly for the vertices of the same polygon. Check also convex polygons.



4. For all vertices.

WORK TO DO: Apply the Rotational Plane Sweep algorithm in all vertices and generate the final visibility graph. You must program a Matlab function with the following input and output parameters:

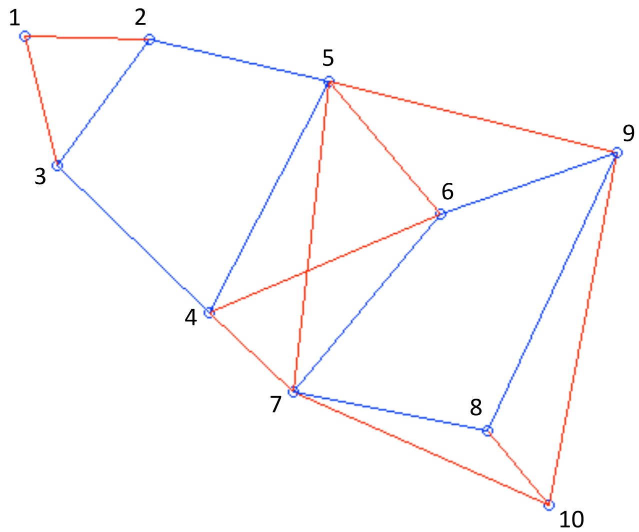
```
function [edges]=RPS(vertices)
```

The “edges” variable contains the list of edges (each one in a different row) indicating the 2 vertex indexes (from the “vertices” variable) that define the edge. When called with the following input variable it must return:

```
>> vertices

vertices =

    0.7807    9.0497         0
    3.0322    8.9912    1.0000
    1.3655    6.7105    1.0000
    4.1140    4.0497    1.0000
    6.2778    8.2310    1.0000
    8.2953    5.8333    2.0000
    5.6345    2.6170    2.0000
    9.1433    1.9152    2.0000
   11.4825    6.9444    2.0000
   10.2544    0.5702    3.0000
```

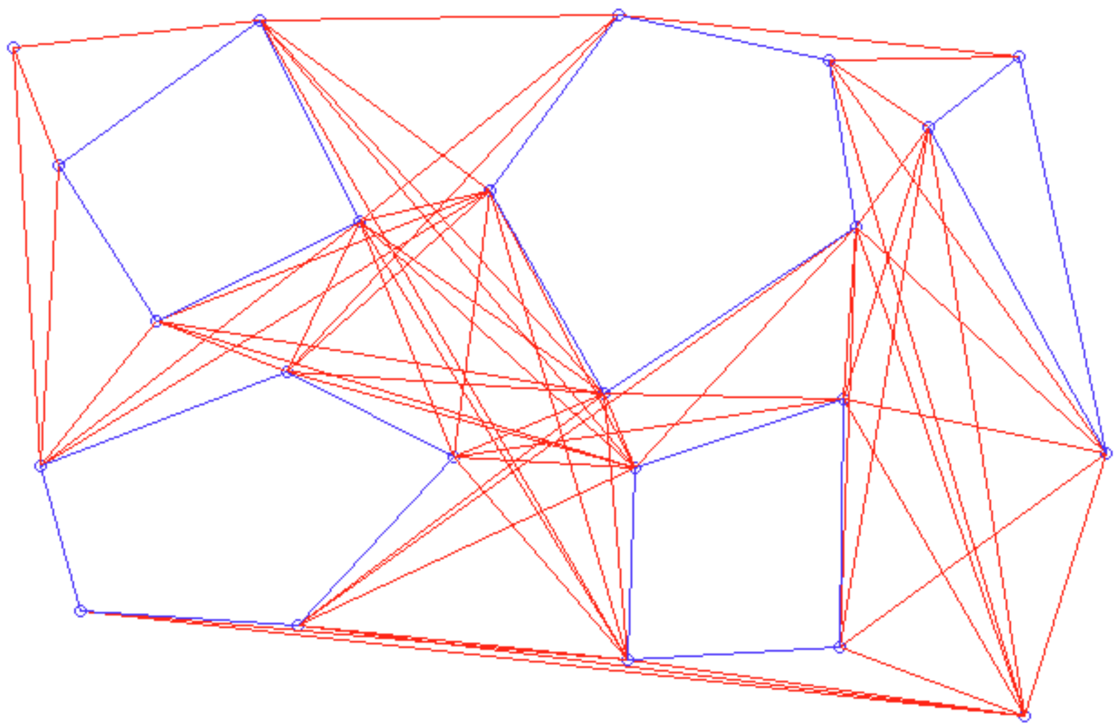
```
>> edges=RPS(vertices)

edges =

     1     3
     1     2
     2     3
     2     5
     3     4
     4     6
     4     5
     4     7
     5     7
     5     6
     5     9
     6     9
     6     7
     7    10
     7     8
     8     9
     8    10
     9    10
```

Note that the edges list will always have the same number of edges, given a particular environment. It is not required to follow any particular order to show the list of edges. Just consider that each edge must appear only once, in a row, containing the 2 vertex indexes from “vertices” variable.



It is very important that you use for the “vertices” and “edges” variables EXACTLY the same format you can see in the previous example. Your “RPS.m” file will be evaluated exclusively based on the results it generates which must follow the previous format. The algorithms will be tested with different environments.

Here you have another example of visibility graph:



and here the 2 corresponding variables:

```
vertices =

    0.6053    7.9971         0
    1.0439    6.8567    1.0000
    2.9737    8.2602    1.0000
    3.9386    6.3304    1.0000
    1.9795    5.3655    1.0000
    6.4532    8.3187    2.0000
    5.1959    6.6228    2.0000
    6.3070    4.6637    2.0000
    8.7339    6.2719    2.0000
    8.4708    7.8801    2.0000
    3.2368    4.8684    3.0000
    0.8684    3.9620    3.0000
    1.2485    2.5585    3.0000
    3.3538    2.4123    3.0000
    4.8450    4.0497    3.0000
    6.5994    3.9327    4.0000
    6.5409    2.0906    4.0000
    8.5877    2.2076    4.0000
    8.6170    4.6053    4.0000
    9.4357    7.2368    5.0000
   11.1608    4.0789    5.0000
   10.3129    7.9094    5.0000
   10.3713    1.5351    6.0000


edges =

     1     3
     1    12
     1     2
     2     3
     2    12
     2     5
     3     6
     3     4
     3    17
     3    16
     3     8
     3     7
     4     7
     4     6
     4     5
     4    12
     4    11
     4    15
     4    17
     4    16
     4     8
     5     7
     5    12
     5    11
     5    16
     5     8
     6    11
     6     7
     6    10
     6    22
     7    12
     7    11
     7    15
     7    17
     7    16
     7     8
     8     9
     8    11
     8    15
     8    14
     8    17
     8    16
     8    19
     9    20
     9    10
     9    14
     9    16
     9    19
     9    18
     9    23
     9    21
    10    22
    10    23
    10    21
    10    20
    11    12
    11    15
    11    16
    12    13
    13    23
    13    17
    13    14
    14    16
    14    15
    14    23
    14    17
    15    19
    15    17
    15    16
    16    19
    16    17
    17    18
    17    23
    18    21
    18    20
    18    19
    18    23
    19    20
    19    23
    19    21
    20    22
    20    23
    20    21
    21    22
    21    23
```

5. Submission.

WORK TO DO: Submit a report in pdf and the Matlab file “RPS.m”. Explain in detail, in the report, the work done in all the sections. Explain also the problems you found. You might want to test your algorithm in other environments.

NOTE that only these 2 files are required and will be evaluated. Do not send more files.