**ROBOT NAVIGATION:**

**Part #1 of Mini-Project:**

|  |  |  |
| --- | --- | --- |
| Gal Gilat | 207766304 | gal.gilat@campus.technion.ac.il |
| Alexander Vasilyev | 337740773 | alksndro@campus.technion.ac.il |
| Ofek Nachshoni | 212594527 | ofekn@campus.technion.ac.il |

Table of Contents:

[Question 1 3](#_Toc202477730)

[Question 2 6](#_Toc202477731)

[Question 3 9](#_Toc202477732)

# Question 1

We computed configuration space (C-space) obstacles for a convex polygonal robot navigating around a single convex polygonal obstacle. We implemented the slice-based method described in Latombe’s algorithm:

1. Defined the robot and obstacle as lists of vertices ordered counterclockwise.
2. For each of 32 regularly spaced orientation values , we:
   * Rotated the robot by
   * Reflected the robot about the origin
   * Computed the Minkowski sum of the reflected robot and the obstacle
   * Extracted the convex hull of the result to represent the obstacle in C-space for
3. Plotted the resulting polygonal C-obstacle slices in the plane for each , and printed the results for slices 1, 8, 16, and 32.

This allowed visualization of how the robot’s configuration constraints change with orientation relative to the obstacle.



figure : Map of Obstacles



figure : Layer 1



figure : Layer 8



figure : Layer 16



figure : Layer 32

# Question 2

In Question 2, we extended the solution from Question 1 to handle a realistic environment with multiple convex polygonal obstacles. The environment consists of outer walls (B01–B04), internal walls (B1–B5), and doors (B6–B7). We used the same algorithm as in Question 1 to compute the configuration space (C-space) obstacle slices for each obstacle individually over 32 discrete rotation angles of the robot.

For each orientation , we computed the Minkowski sum of the rotated robot with every obstacle and extracted the convex hull to obtain the C-obstacle slice. We then overlaid the slices for all obstacles in a single plot for θ-layers 1, 8, 16, and 32. This provided a layered visualization of the full configuration space and demonstrated how the robot’s collision constraints evolve with rotation in a complex environment.



figure : Layer 1



figure : Layer 8



figure : Layer 16



figure : Layer 32

# Question 3

In question 3, we took the printouts of layers 1,8,16,32 and put a “1” value for each cell with the c-obstacles boundaries. All other cells were filled with “0” value. We then plotted each layer and outline the obstacles. Here is the matrices for each layer:

For Layer 1:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 11111111111111111111111111111111 | 11000000000010010111000000000111 | 11000000000010010101000000000111 | 11000000000010010101000000000111 | 11000000000010010101000000000111 | 11000000000010010101000000000111 | 11000000000010010101000000000111 | 11000000000010010111100000000111 | 11000000000010010111100000000111 | 11111111111111111111111111111111 | 01000000000010010111100000000101 | 01000000000010010111100000000101 | 01000000000011110111100000000101 | 01000000000011011111111111111111 | 01000000000011011111100000000111 | 01000000000011011111100000000111 | 01000000000011111010100000000111 | 01000000000001011011100000000111 | 01000000000001011000000000000111 | 01000000000001011011100000000111 | 01000000000001011010100000000111 | 01000000000001011111100000000111 | 01000000000001011111111111111111 | 01000000000001010111100000000101 | 01000000000001010111100000000101 | 01000000000001010111100000000101 | 01000000000001010111100000000101 | 01000000000001010111100000000101 | 01000000000001010111100000000101 | 01000000000001010111100000000101 | 11111111111111111111111111111111 | 11000000000001010101000000000111 |

For Layer 8:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 01000000111100010111000001111011 | 01000111000001111001001110001111 | 01000100001110010001001011110001 | 11111111111111111111111100000001 | 01000100001000010001001000000001 | 01000100001000010001001000000001 | 01000100001000010011101000000001 | 01000100001000111111101000000001 | 01000100001111111000101000000001 | 01000100011110000011101000000001 | 01000111100100111100001000000001 | 01000000000111110000001000000001 | 01000000011110011111111111111111 | 01000011100011110000001000000011 | 01000010011100010000001000001111 | 01000010010000010000001011110001 | 01000010011111111111111100000001 | 01000010000000010000001000000001 | 01000010000000010011101000000001 | 01000010000000111100101000000001 | 01000010000111010111101000000001 | 01000010000101111011101000000001 | 01000010001110111101001000000001 | 01000010001111010001001000000001 | 01000010001000010001001000000001 | 01000010001000010001001000000001 | 01000010001000010001001000000001 | 01000010001000010001001000000001 | 01000010001000010001001000000001 | 11111111111111111111111111111111 | 01000010001000010001001000000011 | 11000010001001110111001000001111 |

For Layer 16:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 01100000000100001101100000001001 | 01111111111111111111111111111111 | 00100000000010001100100000001000 | 00100000000010001110100000001000 | 00100000000011101110100000001000 | 00100000000010111111111111111100 | 00100000000011111010110000001100 | 00100000000011011011110000001100 | 00100000000011111001010000001010 | 00100000000010011001110000001010 | 00100000000010011000000000001010 | 00100000000010001111000000001010 | 00100000000010001101000000001001 | 00100000000010001111000000001001 | 00100000000010001111111111111111 | 00100000000010001010100000001000 | 00100000000010001011100000001000 | 00100000000010001011100000001000 | 00100000000010001011010000001000 | 00100000000010001001010000001000 | 00100000000010001001110000001000 | 00100000000010001001110000001000 | 11111111111111111111111111111100 | 00100000000010001000100000001100 | 00100000000010001000100000001100 | 10100000000001001100100000000110 | 10100000000001001100100000000110 | 10100000000001001100100000000110 | 10100000000001001100100000000110 | 11100000000000101010100000000011 | 11100000000000101010100000000011 | 11111111111111111111111111111111 |

For Layer 32:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 11111111111111111111111111111111 | 11000000000010010111000000000111 | 11000000000010010101000000000111 | 01100000000010001101000000000101 | 01100000000010001100100000000101 | 01100000000010001100100000000101 | 01100000000010001100100000000101 | 00110000000010000111100000000100 | 00110000000010000110110000000100 | 00111111111111111111111111111111 | 00010000000001000110010000000100 | 00010000000001000111010000000100 | 00010000000001110111010000000100 | 00010000000001011111111111111110 | 00010000000001111101011000000110 | 00010000000001101101111000000110 | 00010000000001111100101000000101 | 00010000000001001100111000000101 | 00010000000001001100000000000101 | 00010000000001000111100000000101 | 00010000000001000110100000000100 | 00010000000001000111100000000100 | 00010000000001000111111111111111 | 00010000000001000101010000000100 | 00010000000001000101110000000100 | 00010000000001000101110000000100 | 00010000000001000101101000000100 | 00010000000001000100101000000100 | 00010000000001000100111000000100 | 00010000000001000100111000000100 | 11111111111111111111111111111110 | 10010000000001000100010000000110 |

Here are the grids we got for each layer:



figure : Layer 1



figure : Layer 8



figure : Layer 16



figure : Layer 32

After seeing those results, we thought it is a bit flawed- because it gives room for the robot to be inside the obstacles, so in our opinion it’s a better approach to also fill the inside of the obstacles and not just their boundaries- thus remaining the true free space for the robot to move in. The results we got:  


figure : Layer 1



figure : Layer 8



figure : Layer 16



figure : Layer 32

The code we used in Q1 as requested:

close all; clear all; clc;

import myfunctions.\*

%% definitions

A = [0 0; 8 0; 8 1; 0 1];

B01 = [0 30; 31 30; 31 31; 0 31];

B02 = [0 1; 1 1; 1 30; 0 30];

B03 = [0 0; 31 0; 31 1; 0 1];

B04 = [30 1; 31 1; 31 30; 30 30];

B1 = [0 18; 10 18; 10 19; 0 19];

B2 = [17 17; 18 17; 18 30; 17 30];

B3 = [24 18; 30 18; 30 19; 24 19];

B4 = [0 14; 19 14; 19 15; 0 15];

B5 = [23 13; 31 13; 31 15; 23 15];

B6 = [10 19; 12 19; 12 20; 10 20];

B7 = [22 19; 24 19; 24 20; 22 20];

B\_names = {'B01', 'B02', 'B03', 'B04', 'B1', 'B2', 'B3', 'B4', 'B5', 'B6', 'B7'};

B\_list = {B01, B02, B03, B04, B1, B2, B3, B4, B5, B6, B7};

figure; axis equal; grid minor; hold on;

xlabel('X'); ylabel('Y');

title('Map of Obstacles');

for idx = 1:length(B\_list)

B = B\_list{idx};

fill(B(:,1), B(:,2), 'c', 'FaceAlpha', 0.3, 'EdgeColor', 'b');

text(mean(B(:,1)), mean(B(:,2)), B\_names{idx}, 'HorizontalAlignment', 'center', 'FontSize', 12);

end

hold off;

%% Q1

obstacle\_list = {B01};

obstacle\_name = {'B01'};

slices = cspace\_slices\_multiple(A, obstacle\_list);

layers\_to\_plot = [1, 8, 16, 32]; % Change as needed

for layer = layers\_to\_plot

figure; axis equal; grid minor; hold on;

title(sprintf('C-obstacle slice θ Layer %d', layer));

for idx = 1:length(obstacle\_list)

obstacle = obstacle\_list{idx};

slice\_points = slices{layer, idx};

fill(slice\_points(:,1), slice\_points(:,2), 'r', 'FaceAlpha', 0.3);

text(mean(obstacle(:,1)), mean(obstacle(:,2)), obstacle\_name{idx}, 'HorizontalAlignment', 'center', 'FontSize', 12);

end

hold off;

end

function all\_slices = cspace\_slices\_multiple(A, B\_list)

% A: Nx2 vertices of robot A

% B\_list: cell array of Mx2 obstacles

theta\_values = linspace(0, 2\*pi - 2\*pi/32, 32);

all\_slices = cell(32, length(B\_list));

for k = 1:32

theta = theta\_values(k);

R = [cos(theta), -sin(theta); sin(theta), cos(theta)];

rotated\_A = (R \* A')';

for idx = 1:length(B\_list)

B = B\_list{idx};

B\_col = B'; % 2xM

diff\_points = [];

for i = 1:size(B\_col,2)

for j = 1:size(rotated\_A,1)

diff\_points = [diff\_points, B\_col(:,i) - rotated\_A(j,:)'];

end

end

points = diff\_points';

K = convhull(points(:,1), points(:,2));

all\_slices{k, idx} = points(K, :);

end

end

end