Artificial Intelligence 2019
Problem Sheet 4

Andreas Birk

Notes

The homework serves as preparation for the exams. It is strongly recommended that you solve them before the given deadline - but you do not need to hand them in. Feel free to work on the problems as a group - this is even recommended.

1 Problem

Given the visibility graph from Fig.1. Which edges have to be removed to get a reduced visibility graph?

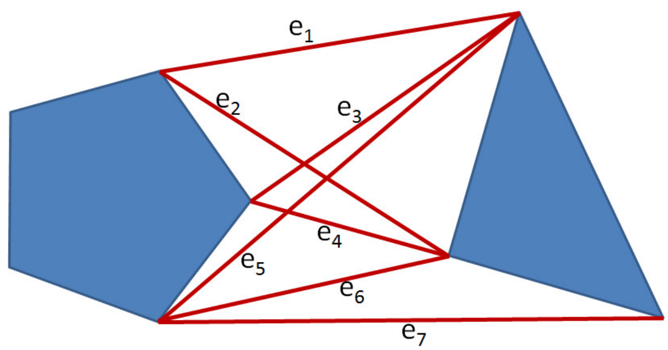

Figure 1: A visibility graph. Find the corresponding reduced visibility graph.

2 Problem

Given the 2D grid map from Fig.2. The map M is a 10×10 array where $M[x][y] = 1$ corresponds to an occupied cell (in addition marked in grey), i.e., an obstacle at (x, y), and $M[x][y] = 0$ indicates free space. Note that the display of the map follows the typical computer graphics convention of placing the origin at the upper left corner and the y-axis pointing downward.

A path P in this map can be formalized by a sequence of points $p_i = (x_i, y_i)$ where each p_{i+1} is in the 4-neighborhood of p_1. A possible path from (2,0) to (3,2) is for example ((2,0),(3,0),(3,1),(3,2)).

Think of A^* with the L_1-norm, i.e., the Manhattan distance to the goal as heuristic to compute the shortest path from p_0 to p_n. When executing A^*, each 4-neighborhood of $p_i = (x_i, y_i)$ is visited in a counter-clockwise manner, starting from the "top", i.e., in the order $(x_i, y_i - 1), (x_i - 1, y_i), (x_i, y_i + 1), (x_i + 1, y_i)$. Suppose the start is (2,0) and the goal is (8,1). What is the intermediate result of A^* after 5 steps, i.e., the candidate shortest path after 5 steps of A^*? What is the final result of A^*?

	0	1	2	3	4	5	6	7	8	9	10
0	1	1	S	0	1	1	1	1	1	1	1
1	1	0	0	0	0	1	0	0	G	0	1
2	1	0	0	0	0	1	0	0	0	0	1
3	1	0	0	0	1	1	1	0	0	0	1
4	1	0	0	0	0	0	0	0	0	0	1
5	1	0	0	0	0	0	0	0	0	0	1
6	1	1	1	1	1	1	1	1	1	1	1

Figure 2: A grid map on which A* should be used for path-planning.

3 Problem

A roadmap based on the Voronoi Diagram (VD) has the advantage that its has a maximum clearance to obstacles. For grid maps and the VD based on the Manhattan Distance, this can be computed using the grassfire transform aka wavefront algorithm, which is sketched in Fig.3. Concretely, the algorithm computes for each cell $D[x][y]$ the Manhattan Distance to the nearest occupied cell $M[x'][y']$.

// initialize all occupied cells with 0, all unoccupied ones with infinity
$\forall$ (x, y) with $M[x][y] = 1$: $D[x][y] = 0$
$\forall$ (x, y) with $M[x][y] = 0$: $D[x][y] = \infty$
// put all occupied cells into queue Q
$\forall$ (x, y) with $M[x][y] = 1$: queue(Q,(x, y))
// take element from Q, update its distance, queue its neighbors if not yet visited
while(notempty(Q) {
    (x, y) = dequeue(Q)
    visit(x, y) = true
    $D[x][y] = \min\ (D[x][y],\ D[x+1][y]{+}1,\ D[x-1][y]{+}1,\ D[x][y+1]{+}1,\ D[x][y-1]{+}1)$
    if($\neg$ visit$(x+1, y)$): queue(Q,$(x+1, y)$)
    if($\neg$ visit$(x-1, y)$): queue(Q,$(x-1, y)$)
    if($\neg$ visit$(x, y+1)$): queue(Q,$(x, y+1)$)
    if($\neg$ visit$(x, y-1)$): queue(Q,$(x, y-1)$)
}

Figure 3: Using the wavefront algorithm to compute a Voronoi based roadmap.

What does D look like when computing it for the map M shown in Fig.4?

4 Problem

To use the result D of the algorithm in Fig.2 as a roadmap, one has to generate a proper Voronoi Diagram (VD). This VD is a graph (V_{VD}, E_{VD}) that can be represented as an array $VD[x][y]$ where $VD[x][y] = 1$ iff (x, y) is a vertex in VD, i.e., $(x, y) \in V_{VD}$. $VD[x][y]$ can simply be set to zero otherwise, i.e., if $(x, y) \notin V_{VD}$.

There is an edge between the two VD vertices (x, y) and (x', y'), i.e., $((x, y), (x', y')) \in E_{VD}$ iff they are neighbors in a 8-neighborhood, i.e., $((x, y), (x', y')) \in E_{VD}$ iff $x = x' \pm 1$ or $y = y' \pm 1$. Note that these

	0	1	2	3	4	5	6	7	8	9
0	1	1	0	0	1	1	1	1	1	1
1	1	0	0	0	0	1	0	0	0	1
2	1	0	0	0	0	1	0	0	0	1
3	1	0	0	0	0	1	0	0	0	1
4	1	0	0	0	0	1	0	0	0	1
5	1	0	0	0	0	1	0	0	0	1
6	1	0	0	1	1	1	1	0	0	1
7	1	0	0	0	0	0	0	0	0	1
8	1	0	0	0	0	0	0	0	0	1
9	1	1	1	1	1	1	1	1	1	1

Figure 4: A grid map on which a VD should be computed.

edges are implicitly defined once the VD vertices are computed. To compute the VD vertices, one has to extract the locations (x, y) that are local maximums in D in a 4-neighborhood, i.e., $VD[x][y] = 1$ iff $D(x, y) = max$(D[x][y],D[x+1][y],D[x-1][y],D[x][y+1],D[x][y-1]).

What does the VD of the map M from Fig.2 look like? What is the shortest path from (2,0) to (8,1) that A^* generates on this VD?

5 Problem

Given a regular grid map M of size !0×!0 with $M[x, y] = 0$ for free space and $M[x, y] = 1$ for occupied space. Suppose there are only two simple point obstacles $M[5, 4] = 1$ and $M[9, 6] = 1$. The goal point g is at $(7, 8)$, the robot is at $(1, 2)$. Furthermore, following potential field functions are given:

- Attractive Field $U_a()$ for a goal at g:
  $U_a(p) = c_a(d(p, g))^2$
- Repulsive Field $U_r()$ for an obstacle at o:
  $U_r(p) = c_r(1/d(p, o) - 1/c_d)^2$ if $d(p, o) \leq c_d$, otherwise $U_r(p) = 0$

with $d()$ is the Euclidean Distance, distance parameter c_d for repulsive fields to take effect is $c_d = 5$, and the scaling parameters are $c_a = c_r = 1$.

What does the total *Force* that acts on the robot? What does the Field look like in the 8-neighborhood of the current robot location? Note: Take care to properly distinguish between the potential field function and the force field.