Potential Fields

Robots in Context

Leon Bodenhagen lebo@mmmi.sdu.dk

Associate Professor

SDU Robotics, Maersk Mc-Kinney Moller Institute

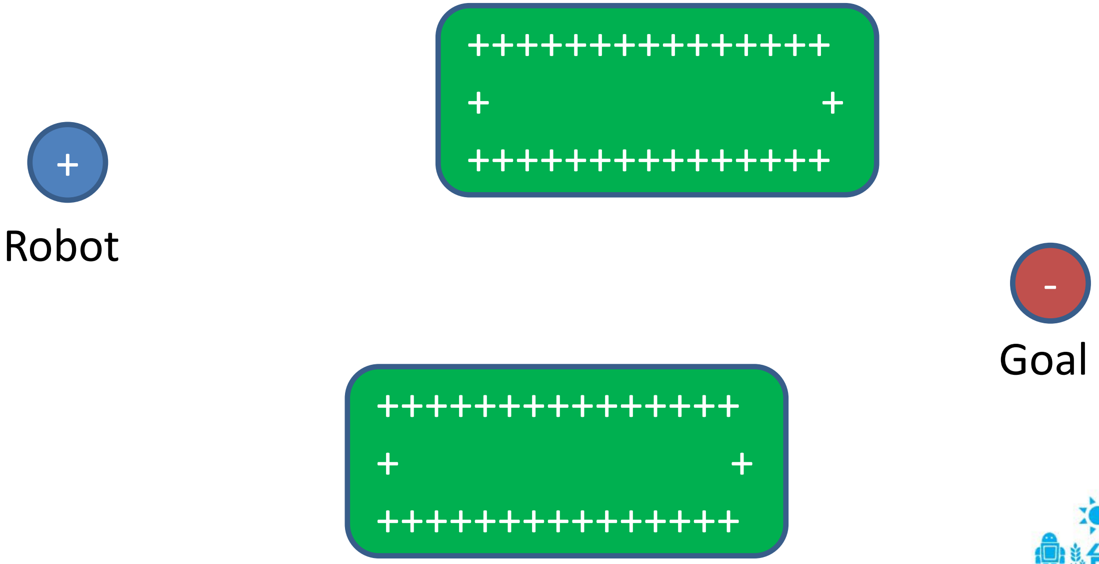
Content

- Potential functions
  - Attractive and Repulsive potentials
  - Gradient descent
  - Brushfire
- The Wavefront planner

- Last part:
  - Roadmaps

Potential Functions

- Basic idea
  - The goal is attracting the robot
  - Objects are repelling the robot



”Simple” Potential Functions

- Attractive Potential
  - $U_{att}(q) = \frac{1}{2}\gamma\, d(q, q_{goal})^2$
  - Where $d(q, q_{goal})$ is a distance function and γ determines the strength of the attractive potential

”Simple” Potential Functions

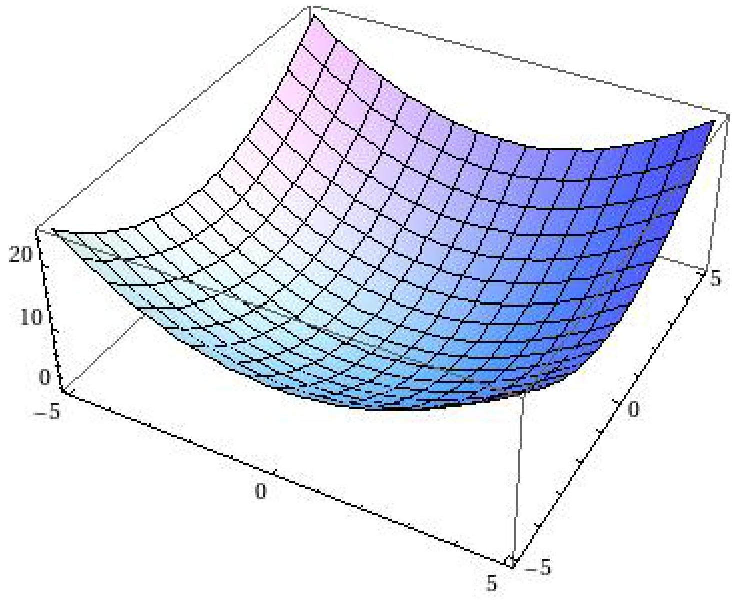
- Attractive Potential
  - $U_{att}(q) = \frac{1}{2}\gamma\, d(q, q_{goal})^2$
  - Where $d(q, q_{goal})$ is a distance function and γ determines the strength of the attractive potential
- Repulsive Potential
  - $U_{rep}(q) = \begin{cases} \frac{1}{2}\mu\,(\frac{1}{D(q)} - \frac{1}{Q^*})^2, & D(q) \le Q^* \\ 0, & D(q) > Q^* \end{cases}$
  - Where $D(q)$ is the distance to the nearest obstacle, μ determines the strength of the repulsive potential and Q^* is a distance threshold.

Exercise: Draw the potentials assuming q is 2-dimensional

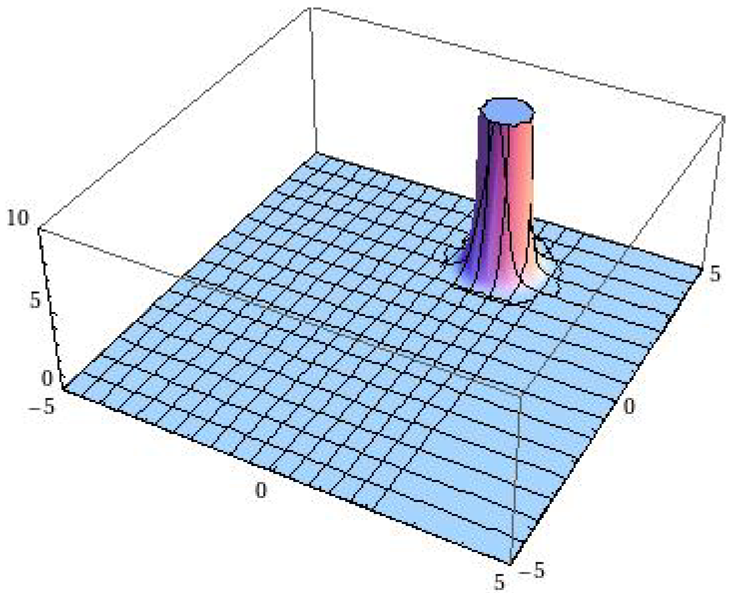
- Attractive Potential
  - $U_{att}(q) = \frac{1}{2}\gamma\, d(q, q_{goal})^2$
  - Where $d(q, q_{goal})$ is a distance function and γ determines the strength of the attractive potential
- Repulsive Potential
  - $U_{rep}(q) = \begin{cases} \frac{1}{2}\mu\,(\frac{1}{D(q)} - \frac{1}{Q^*})^2, & D(q) \leq Q^* \\ 0, & D(q) > Q^* \end{cases}$
  - Where $D(q)$ is the distance to the nearest obstacle, μ determines the strength of the repulsive potential and Q^* is a distance threshold.

Exercise: Draw the potentials assuming q is 2-dimensional

- Attractive Potential



- Repulsive



Choosing reasonable parameters, the previous expressions can be visualized directly

Combining attractive and repulsive potentials

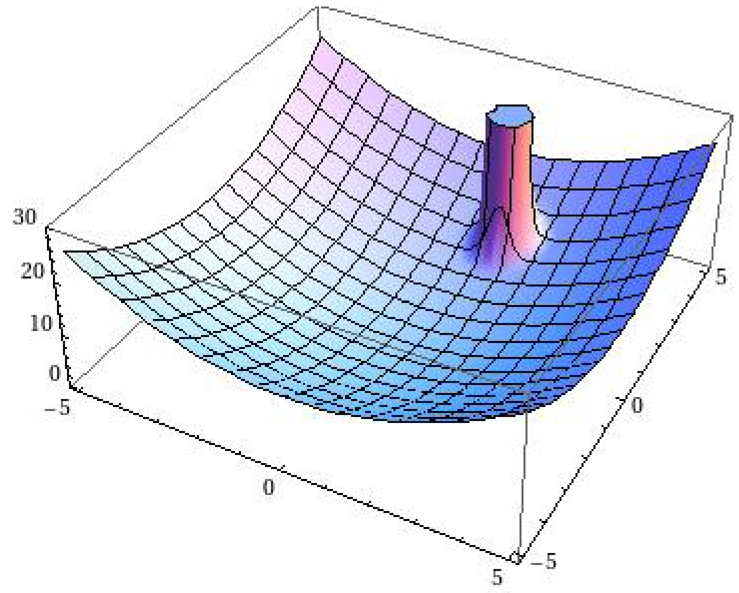
- Add the two potentials
  - $U(q) = U_{att}(q) + U_{rep}(q)$
- In case of multiple obstacles we can either use
  - $U(q) = U_{att}(q) + \sum_{i=1}^{N} U_{rep,i}(q)$

  or

  - $U(q) = U_{att}(q) + \min_{i} U_{rep}(q)$

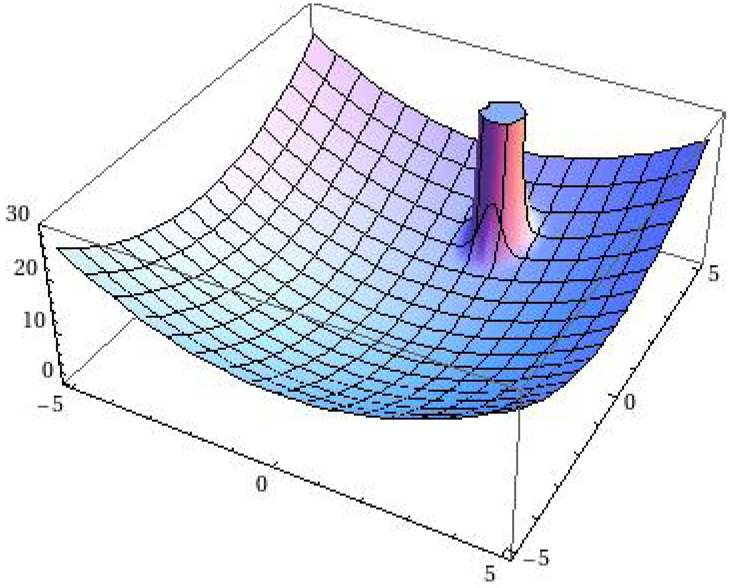
Combining attractive and repulsive potentials

- Combined attractive and repulsive potential



Using the Potential Field

- Idea: Follow the slope downwards until reaching the goal



- Gradient Descent

$$q(0) = q_{start}$$
$$i = 1$$
$$\text{while } (\nabla U(q(i)) \neq 0)$$
$$\quad q(i+1) = q(i) - \alpha(i)\,\nabla U(q(i))$$
$$\quad i = i + 1$$

Gradients

What is the gradient of the attractive potential

$$U_{att}(q) = \frac{1}{2}\gamma\, d(q, q_{goal})^2$$

What is the gradient of the repulsive potential

$$U_{rep}(q) = \begin{cases} \frac{1}{2}\mu\left(\frac{1}{D(q)} - \frac{1}{Q^*}\right)^2, & D(q) \le Q^* \\ 0, & D(q) > Q^* \end{cases}$$

Gradients

What is the gradient of the attractive potential $U_{att}(q) = \frac{1}{2}\gamma\, d(q, q_{goal})^2$

$\nabla U_{att}(q) = \frac{1}{2}\gamma\, \nabla d(q, q_{goal})^2 = \gamma\,(q - q_{goal})$

Gradients

What is the gradient of the attractive potential $U_{att}(q) = \frac{1}{2}\gamma\, d(q, q_{goal})^2$

$$\nabla U_{att}(q) = \frac{1}{2}\gamma\, \nabla d(q, q_{goal})^2 = \gamma\,(q - q_{goal})$$

What is the gradient of the repulsive potential $U_{rep}(q) = \begin{cases} \frac{1}{2}\mu\,(\frac{1}{D(q)} - \frac{1}{Q^*})^2, & D(q) \leq Q^* \\ 0, & D(q) > Q^* \end{cases}$

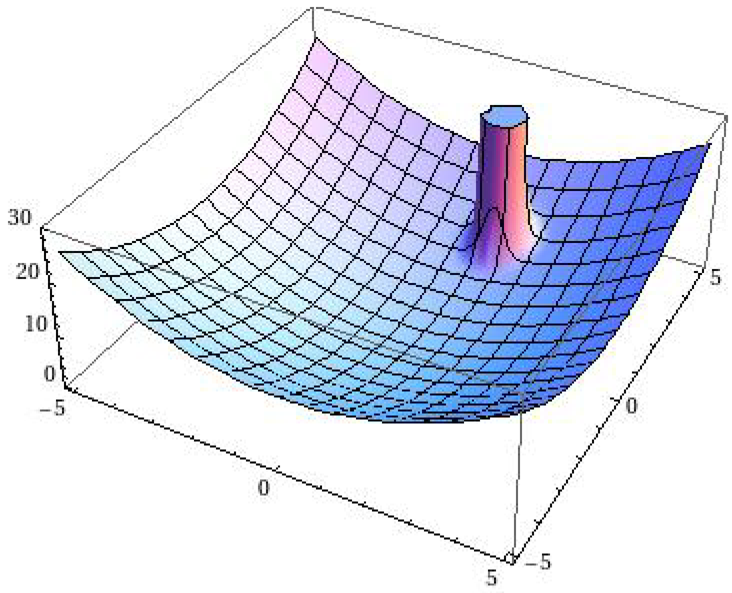
$$\nabla U_{rep}(q) = \begin{cases} \mu(\frac{1}{Q^*} - \frac{1}{D(q)})\,\frac{1}{D(q)^2}\nabla D(q), & D(q) \leq Q^* \\ 0, & D(q) > Q^* \end{cases}$$

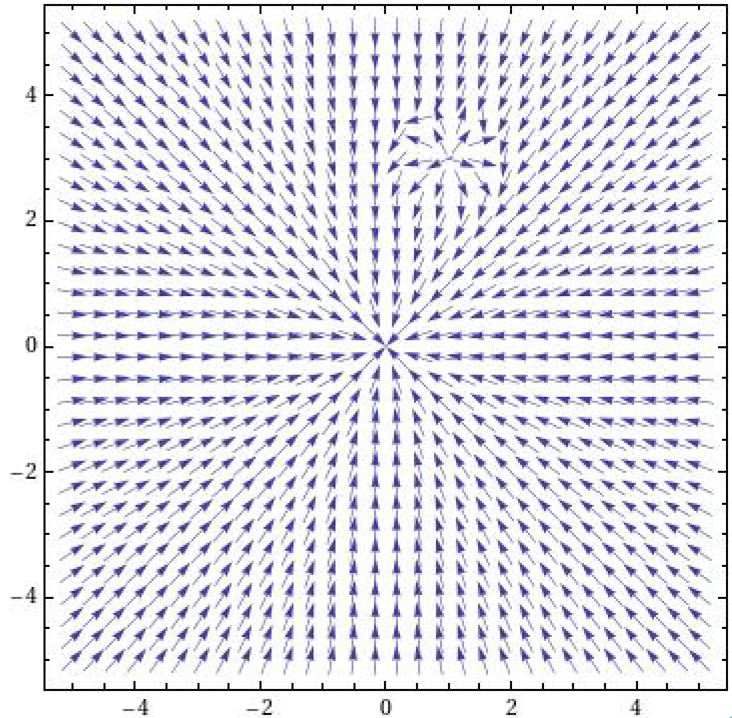
Gradients

- What is the physical interpretation of a first order derivative of a potential?

Gradient Combined

- What is the physical interpretation of a first order derivative of a potential?



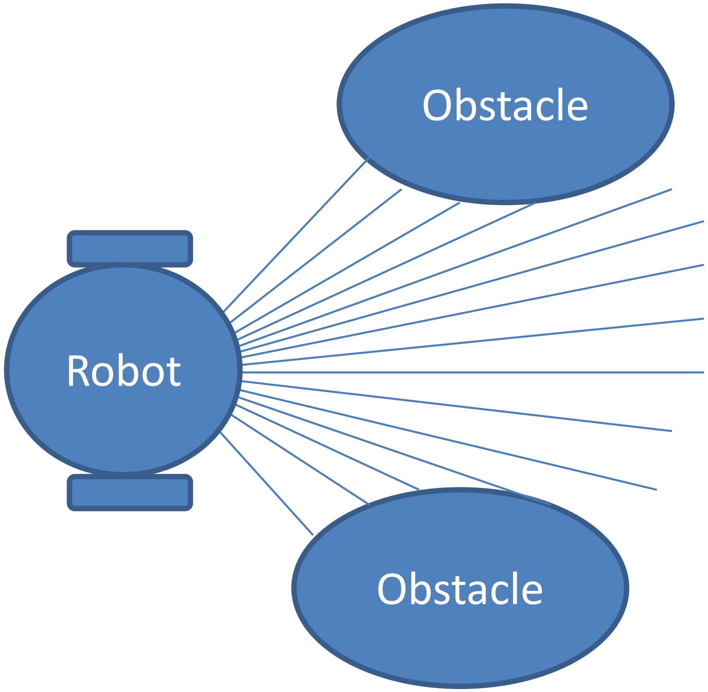


Algorithm Characteristics

- Completeness properties?
  - Any particular scenarious where if and where it might get stuck?
- Online vs. Offline
- Sensor based vs. Model based
- Greedy vs. exhaustive

Sensor Based Potential Field

- Use e.g. laser scanner to detect obstacles.
- Use the closest reading as the obstacle in the potential.
- The gradient of the repulsive potential is given based on the direction of the reading.



Model Based Potential Field
The Brushfire Algorithm

- Assume that we have a regular grid with cells either being free (value 0) or obstacles (value 1).

			1						
			1						
			1		1	1	1		
							1	1	
							1	1	

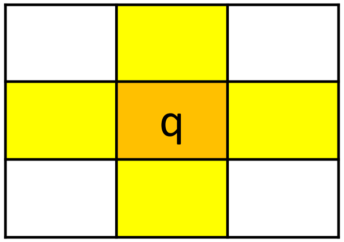
Model Based Potential Field
The Brushfire Algorithm

- We now wish to define the distance function D(q) for each cell

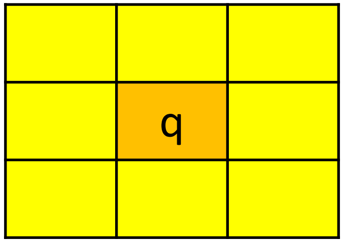
			1						
			1						
			1		1	1	1		
							1	1	
							1	1	

Model Based Potential Field
The Brushfire Algorithm

- Defining the neighbors of a cell
  - 4 Connected: Connect only direct neighbors



  - 8 Connected: Connect also diagonally



Model Based Potential Field
The Brushfire Algorithm

- Start labeling all nodes next to an obstacle 2

		2	2	2					
		2	1	2					
		2	1	2	2	2	2	2	
		2	1	2	1	1	1	2	
		2	2	2	2	2	2	2	2
						2	1	1	2
						2	1	1	2
						2	2	2	2

Model Based Potential Field
The Brushfire Algorithm

- Next label all neighbors to 2’s as 3

	3	2	2	2	3				
	3	2	1	2	3	3	3	3	3
	3	2	1	2	2	2	2	2	3
	3	2	1	2	1	1	1	2	3
	3	2	2	2	2	2	2	2	2
	3	3	3	3	3	2	1	1	2
					3	2	1	1	2
					3	2	2	2	2

Model Based Potential Field
The Brushfire Algorithm

- And so forth…

4	3	2	2	2	3	4	4	4	4
4	3	2	1	2	3	3	3	3	3
4	3	2	1	2	2	2	2	2	3
4	3	2	1	2	1	1	1	2	3
4	3	2	2	2	2	2	2	2	2
4	3	3	3	3	3	2	1	1	2
4	4	4	4	4	3	2	1	1	2
5	5	5	5	4	3	2	2	2	2

Model Based Potential Field
The Brushfire Algorithm

- And so forth…
- What about the border region?

4	3	2	2	2	3	4	4	4	4
4	3	2	1	2	3	3	3	3	3
4	3	2	1	2	2	2	2	2	3
4	3	2	1	2	1	1	1	2	3
4	3	2	2	2	2	2	2	2	2
4	3	3	3	3	3	2	1	1	2
4	4	4	4	4	3	2	1	1	2
5	5	5	5	4	3	2	2	2	2

Model Based Potential Field
The Brushfire Algorithm

- The distance to an obstacle can now be defined based on the value of a cell.
- The gradient as the direction to a neighbor with a smaller value.

4	3	2	2	2	3	4	4	4	4
4	3	2	1	2	3	3	3	3	3
4	3	2	1	2	2	2	2	2	3
4	3	2	1	2	1	1	1	2	3
4	3	2	2	2	2	2	2	2	2
4	3	3	3	3	3	2	1	1	2
4	4	4	4	4	3	2	1	1	2
5	5	5	5	4	3	2	2	2	2

The Wave-front planner

- Apply a brush fire like algorithm from the goal.
- Assume that obstacles have value 1 and free space has value 0.
- Algorithm:
  - Assign the goal cell the value 2
  - Add the neighbors of the goal cell to a set L
  - While L is not empty
    - Take en element of L
    - Assign the element the value of the its neighbor+1
    - Add all neighbors with 0 value to L

Wavefront planner: setup

7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0
3	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Wavefront in action (part 1)

7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0
3	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	3
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	2
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Wavefront in action (part 2)

7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0
3	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	4	4	4
1	0	0	0	0	0	0	0	0	0	0	0	0	0	4	3	3
0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	3	2
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Wavefront in action (part 3)

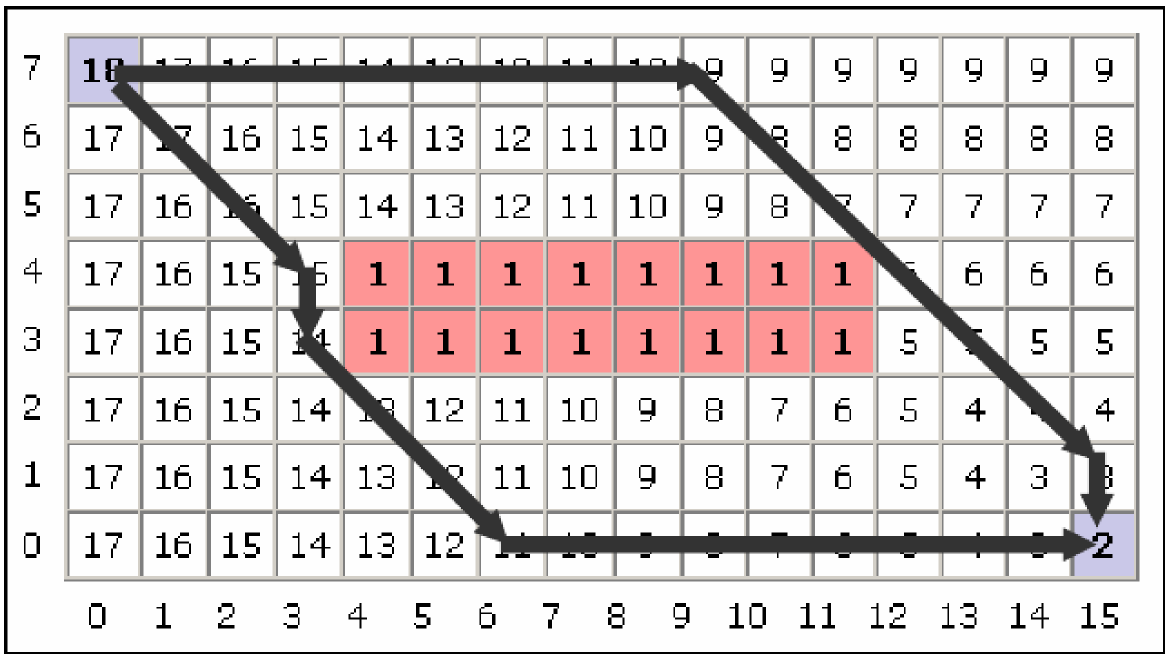
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0
3	0	0	0	0	1	1	1	1	1	1	1	1	5	5	5	5
2	0	0	0	0	0	0	0	0	0	0	0	0	5	4	4	4
1	0	0	0	0	0	0	0	0	0	0	0	0	5	4	3	3
0	0	0	0	0	0	0	0	0	0	0	0	0	5	4	3	2
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Wavefront in action (done)

7	18	17	16	15	14	13	12	11	10	9	9	9	9	9	9	9
6	17	17	16	15	14	13	12	11	10	9	8	8	8	8	8	8
5	17	16	16	15	14	13	12	11	10	9	8	7	7	7	7	7
4	17	16	15	15	1	1	1	1	1	1	1	1	6	6	6	6
3	17	16	15	14	1	1	1	1	1	1	1	1	5	5	5	5
2	17	16	15	14	13	12	11	10	9	8	7	6	5	4	4	4
1	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	3
0	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Wavefront, now what?

- To find the shortest path, according to your metric, simply always move toward a cell with a lower number
  - The numbers generated by the Wavefront planner are roughly proportional to their distance from the goal



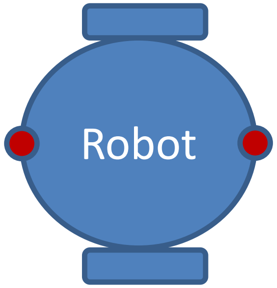
Two possible Shortest paths

Wavefront Properties

- What can be said about the properties of the algorithm?
  - Completeness?
  - Greedy or exhaustive?

Potential function on rigid body

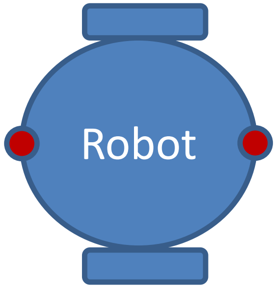
- Select N points (●) on the robot
  - N should be large enough to control the robot AND represent its shape relative to obstacles.



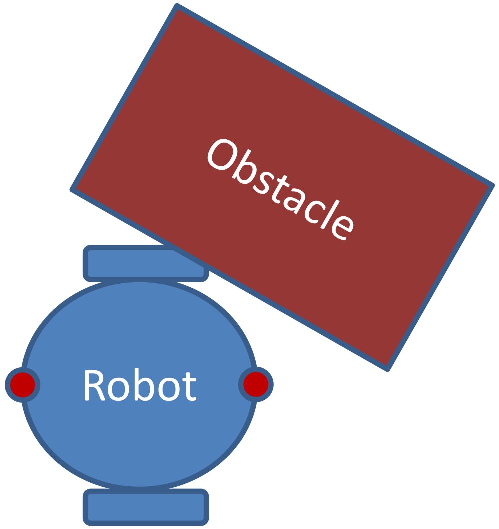
Enough to pin down the robot

Potential function on rigid body

- Select N points (●) on the robot
  - N should be large enough to control the robot AND represent its shape relative to obstacles.



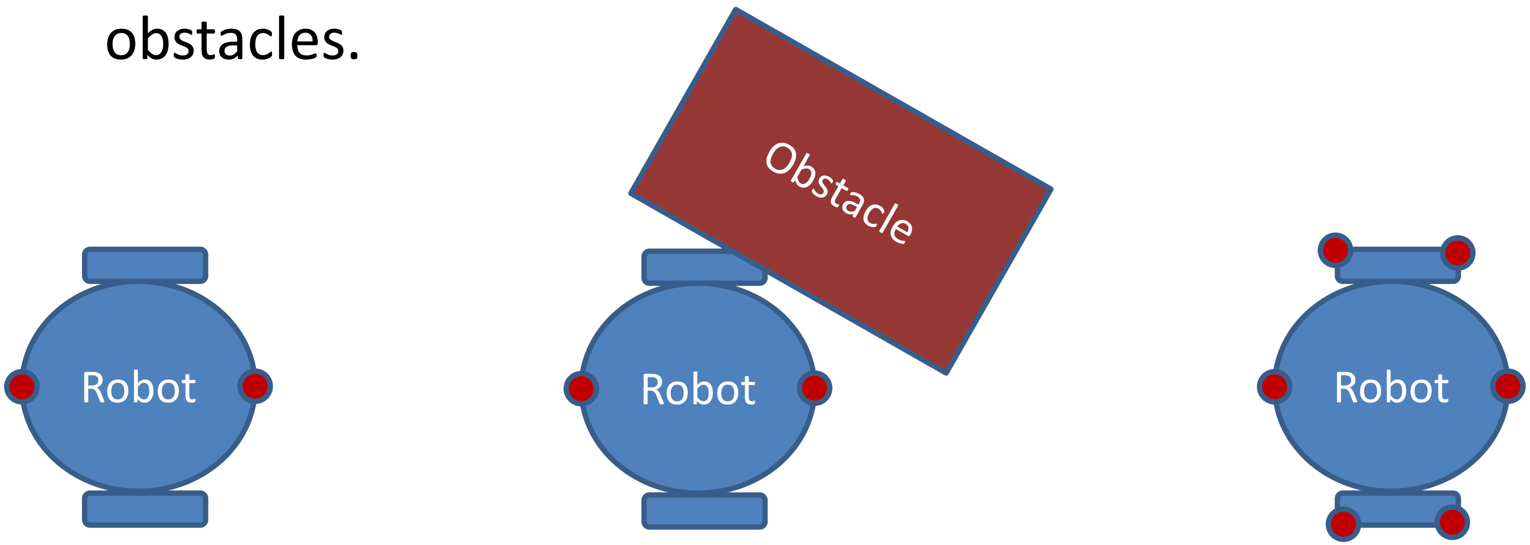
Enough to pin down the robot



Not enough to avoid collision

Potential function on rigid body

- Select N points (●) on the robot
  - N should be large enough to control the robot AND represent its shape relative to obstacles.



Enough to pin down the robot

Not enough to avoid collision

When obstacles are "large" point representing the convex hull will be enough

Potential function on rigid body

- Attractive potential for each point

$$U_{\mathrm{att},j}(q) = \begin{cases} \frac{1}{2}\zeta_i d^2(r_j(q), r_j(q_{\mathrm{goal}})), & d(r_j(q), r_j(q_{\mathrm{goal}})) \le d^*_{\mathrm{goal}} \\ d^*_{\mathrm{goal}}\zeta_j d(r_j(q), r_j(q_{\mathrm{goal}})) - \frac{1}{2}\zeta_j d^*_{\mathrm{goal}}, & d(r_j(q), r_j(q_{\mathrm{goal}})) > d^*_{\mathrm{goal}}. \end{cases}$$

$$\nabla U_{\mathrm{att},j}(q) = \begin{cases} \zeta_i(r_j(q) - r_j(q_{\mathrm{goal}})), & d(r_j(q), r_j(q_{\mathrm{goal}})) \le d^*_{\mathrm{goal}}, \\ \frac{d^*_{\mathrm{goal}}\zeta_j(r_j(q) - r_j(q_{\mathrm{goal}}))}{d(r_j(q), r_j(q_{\mathrm{goal}}))}, & d(r_j(q), r_j(q_{\mathrm{goal}})) > d^*_{\mathrm{goal}}. \end{cases}$$

- Repelling potential for each point

$$U_{\mathrm{rep}i,j}(q) = \begin{cases} \frac{1}{2}\eta_j \left(\frac{1}{d_i(r_j(q))} - \frac{1}{Q_i^*}\right)^2, & d_i(r_j(q)) \le Q_i^*, \\ 0, & d_i(r_j(q)) > Q_i^*, \end{cases}$$

$$\nabla U_{\mathrm{rep}i,j}(q) = \begin{cases} \eta_j \left(\frac{1}{Q_i^*} - \frac{1}{d_i(r_j(q))}\right) \frac{1}{d_i^2(r_j(q))} \nabla d_i(r_j(q)), & d_i(r_j(q)) \le Q_i^*, \\ 0, & d_i(r_j(q)) > Q_i^*. \end{cases}$$

Potential Fields for Manipulator

- Pick several points on the manipulator
- Compute attractive and repulsive potentials for each
- Transform these into the configuration space
- Use the resulting force to move the robot (in its configuration space)