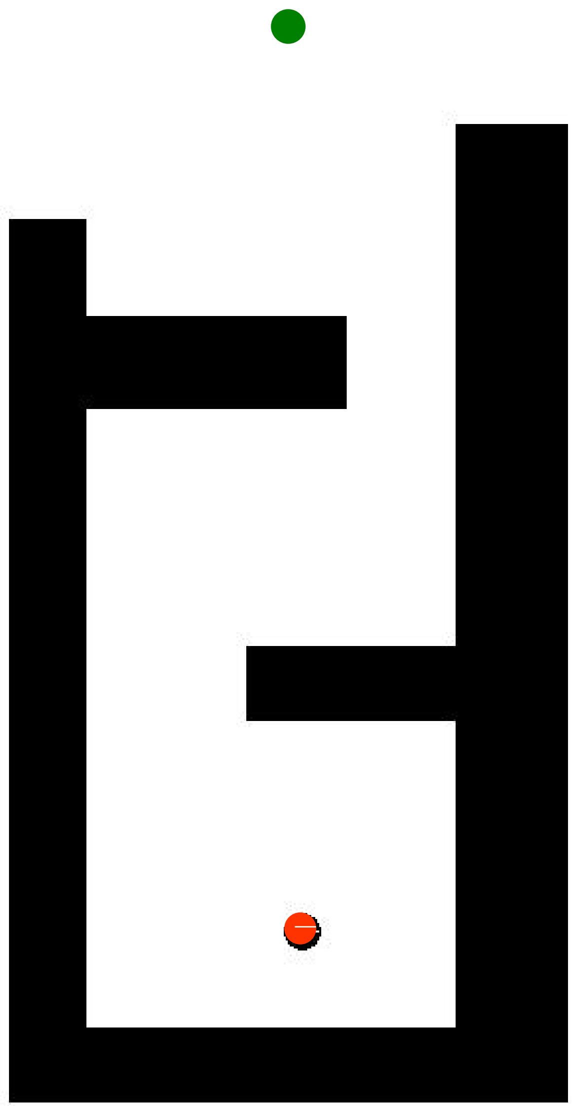
Robotic Motion Planning: Review C-Space and Start Potential Functions

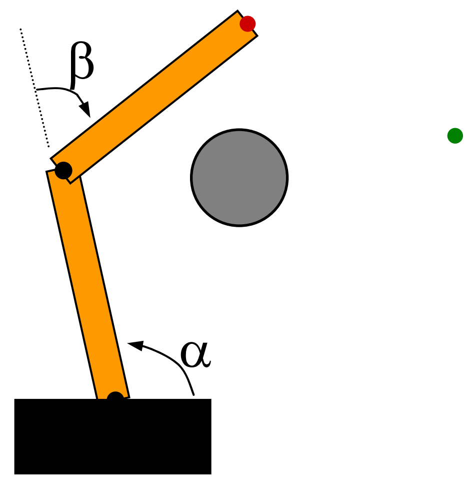
Robotics Institute 16-735
http://voronoi.sbp.ri.cmu.edu/~motion

Howie Choset
http://voronoi.sbp.ri.cmu.edu/~choset

What if the robot is not a point?

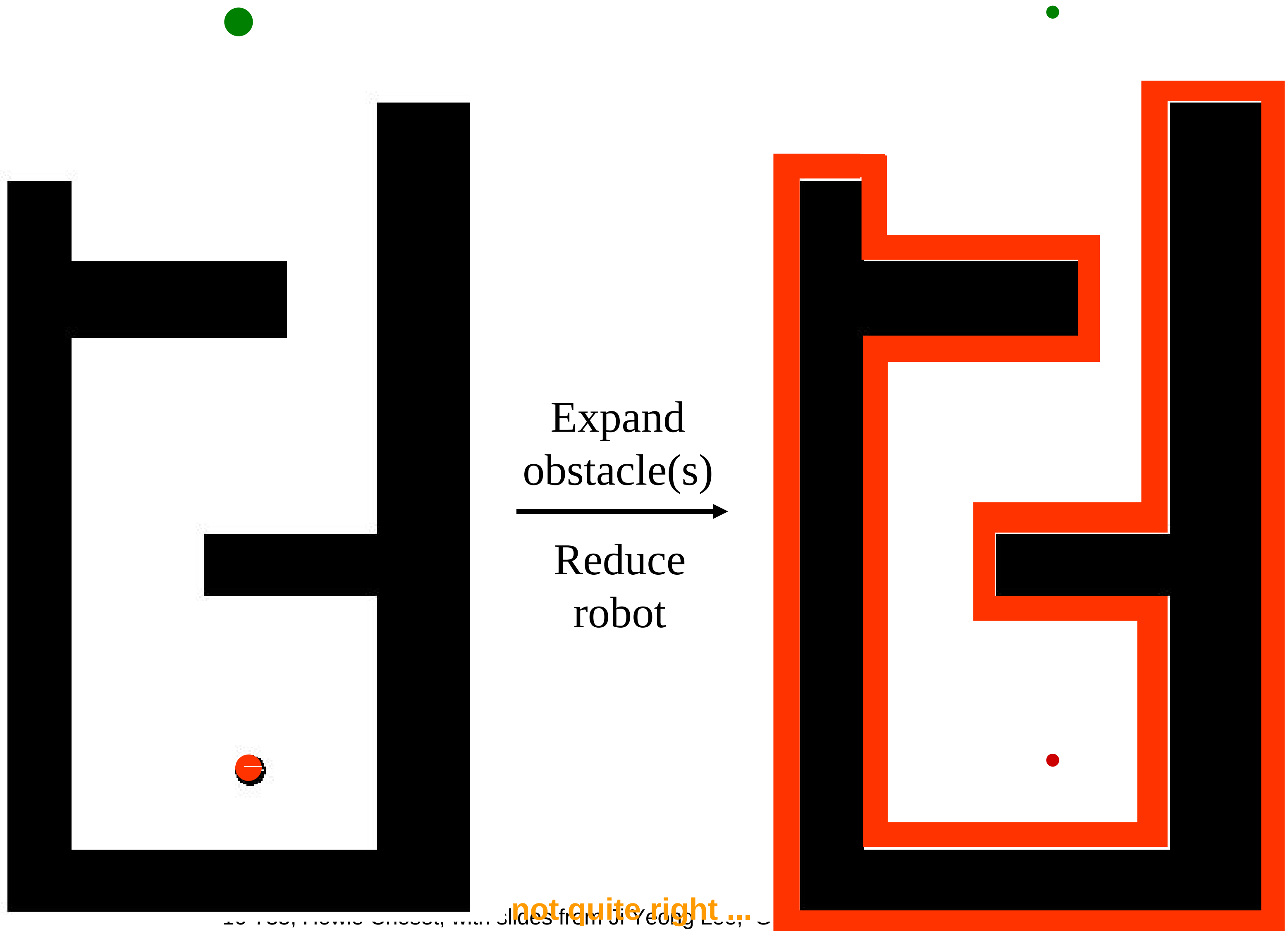


The Scout should probably not be modeled as a point...

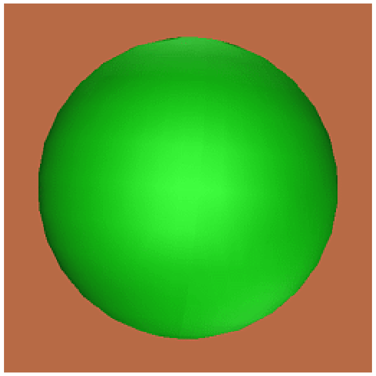


Nor should robots with extended linkages that may contact obstacles...

What is the position of the robot?

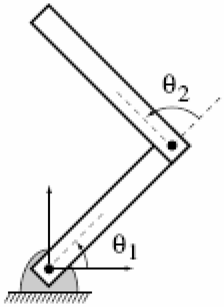


Topology?



Sphere?

Torus?

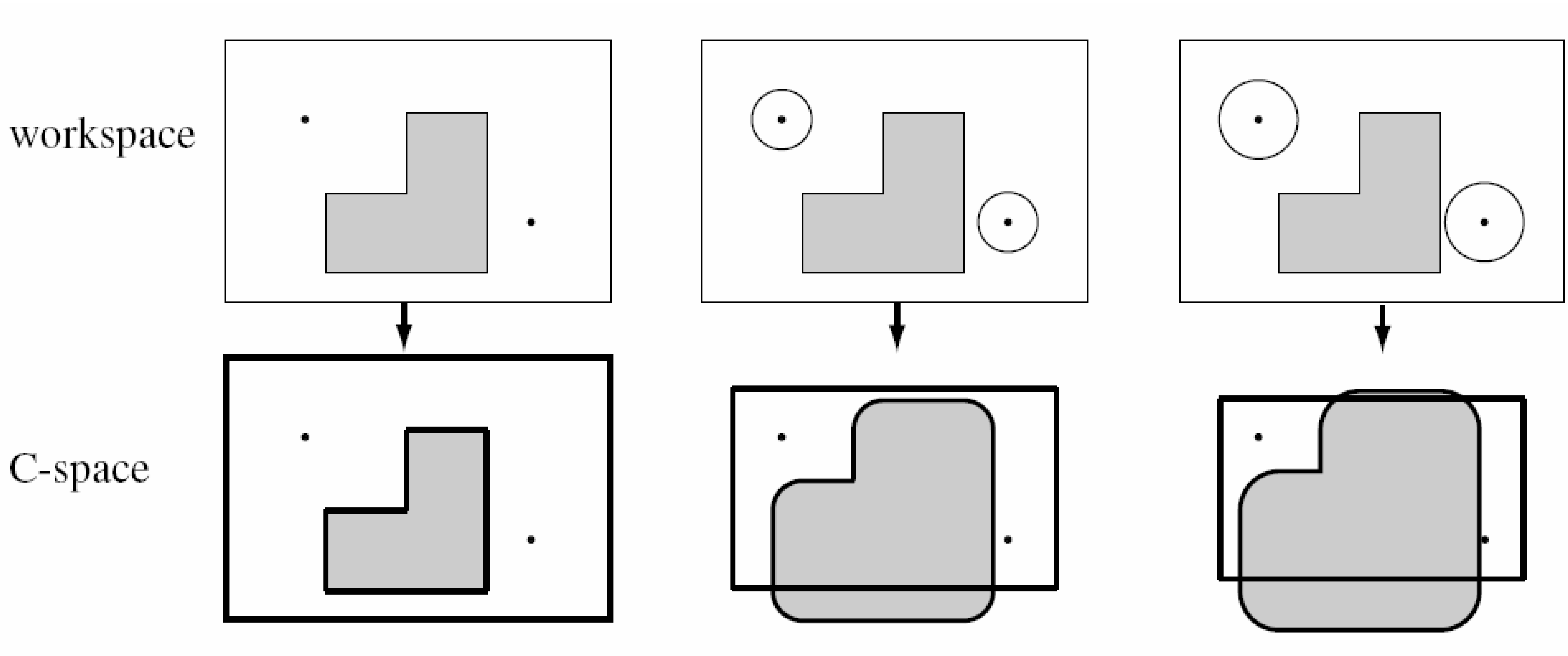


(a)

2R manipulator

Configuration space

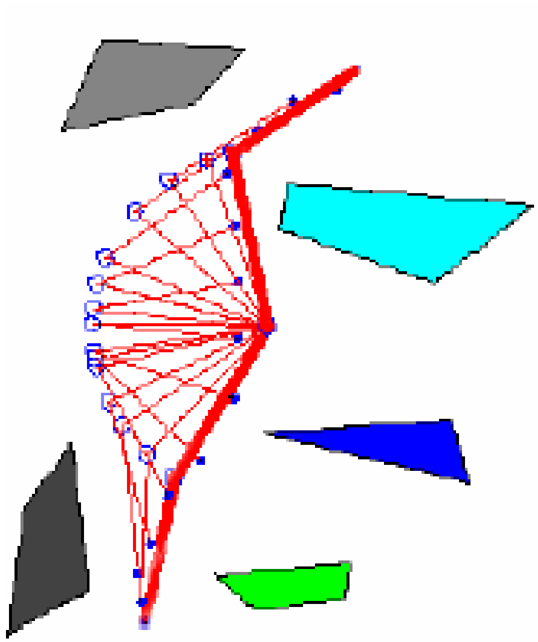
Trace Boundary of Workspace

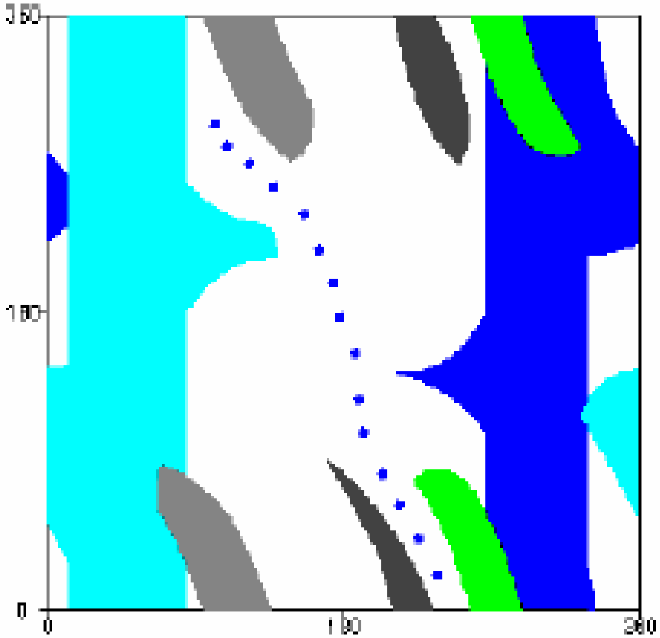


$$\mathcal{QO}_i = \{q \in \mathcal{Q} \mid R(q) \bigcap \mathcal{WO}_i \neq \emptyset\}.$$

Pick a reference point…

Two Link Path



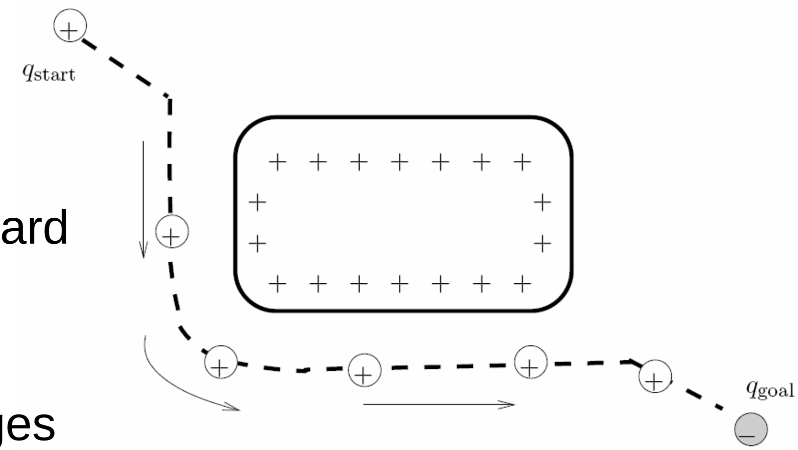


http://ford.ieor.berkeley.edu/cspace/

Thanks to Ken Goldberg

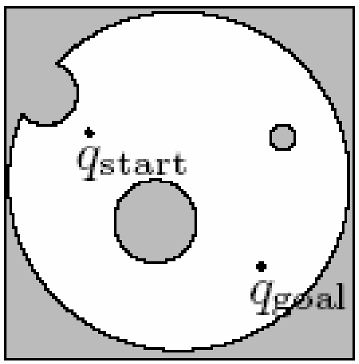
The Basic Idea

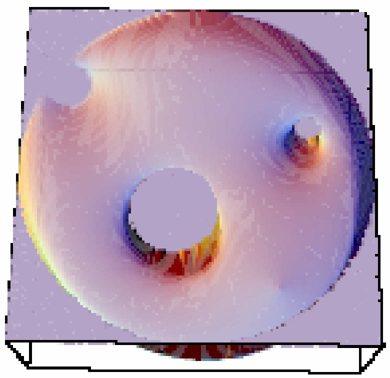
- A really simple idea:
  - Suppose the goal is a point $g \in \Re^2$
  - Suppose the robot is a point $r \in \Re^2$
  - Think of a "spring" drawing the robot toward the goal and away from obstacles:
  - Can also think of like and opposite charges



Another Idea

- Think of the goal as the bottom of a bowl
- The robot is at the rim of the bowl
- What will happen?





The General Idea

- Both the bowl and the spring analogies are ways of storing potential *energy*
- The robot moves to a lower energy configuration
- A *potential function* is a function U : $\Re^m \rightarrow \Re$
- Energy is minimized by following the negative *gradient* of the potential energy function:

$$\nabla U(q) = DU(q)^T = [\frac{\partial U}{\partial q_1}(q), \ldots, \frac{\partial U}{\partial q_m}(q)]^T$$

- We can now think of a *vector field* over the space of all q's ...
  - at every point in time, the robot looks at the vector at the point and goes in that direction

Attractive/Repulsive Potential Field

$$U(q) = U_{\text{att}}(q) + U_{\text{rep}}(q)$$

– U_{att} is the "attractive" potential --- move to the goal

– U_{rep} is the "repulsive" potential --- avoid obstacles

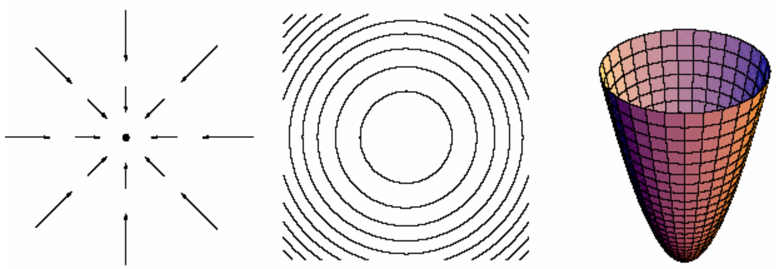
Artificial Potential Field Methods: Attractive Potential

Conical Potential

$$U(q) = \zeta d(q, q_{\text{goal}}).$$

$$\nabla U(q) = \frac{\zeta}{d(q, q_{\text{goal}})}(q - q_{\text{goal}}).$$

Quadratic Potential


(a) (b) (c)

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

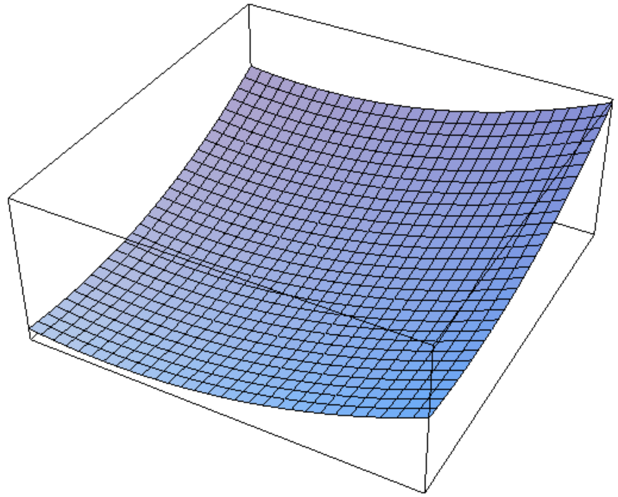
$$\begin{aligned} F_{\text{att}}(q) = \nabla U_{\text{att}}(q) &= \nabla\left(\frac{1}{2}\zeta d^2(q, q_{\text{goal}})\right), \\ &= \frac{1}{2}\zeta \nabla d^2(q, q_{\text{goal}}), \\ &= \zeta(q - q_{\text{goal}}), \end{aligned}$$

Artificial Potential Field Methods: Attractive Potential

Combined Potential

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

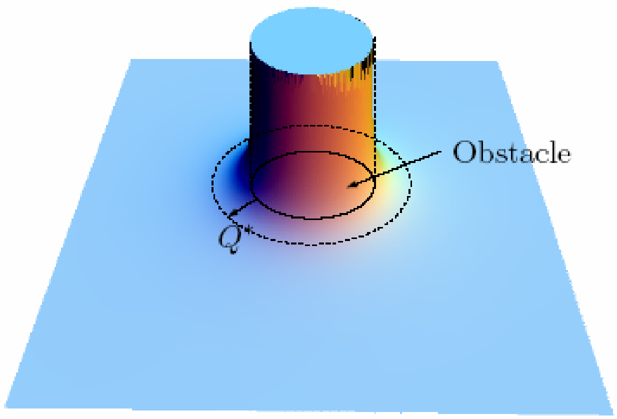
$$\nabla U_{\text{att}}(q) = \begin{cases} \zeta(q - q_{\text{goal}}), & d(q, q_{\text{goal}}) \le d^*_{\text{goal}}, \\ \frac{d^*_{\text{goal}}\zeta(q - q_{\text{goal}})}{d(q, q_{\text{goal}})}, & d(q, q_{\text{goal}}) > d^*_{\text{goal}}, \end{cases}$$



In some cases, it may be desirable to have distance functions that grow more slowly to avoid huge velocities far from the goal

one idea is to use the quadratic potential near the goal (< d*) and the conic farther away
One minor issue: what?

The Repulsive Potential

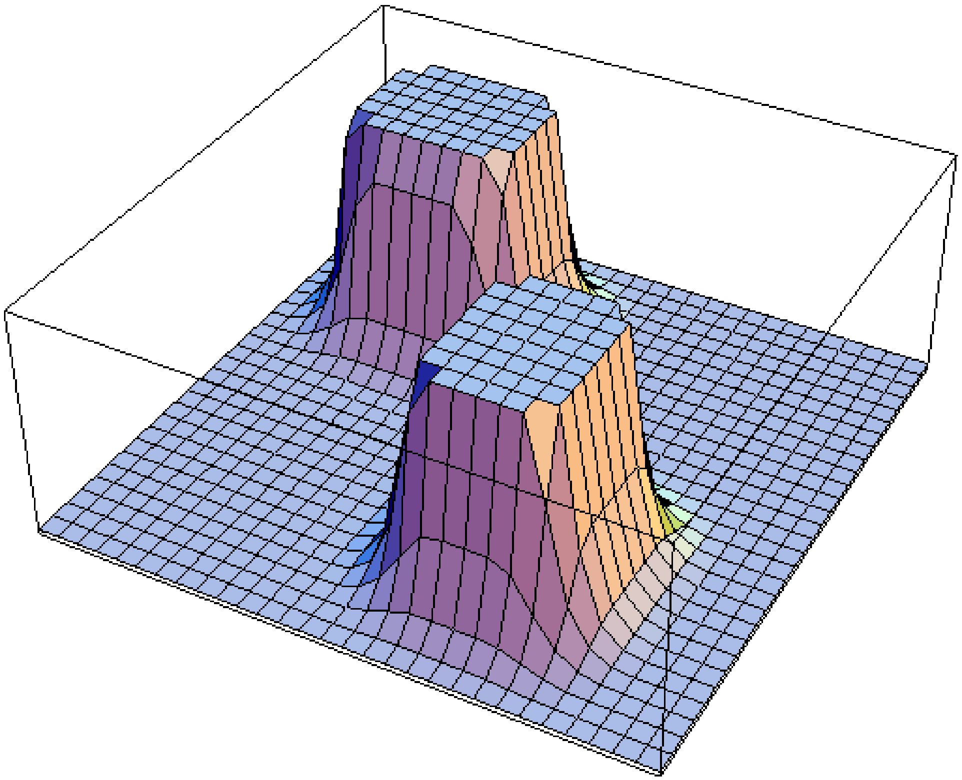


$$U_{\text{rep}}(q) = \begin{cases} \frac{1}{2}\eta\left(\frac{1}{D(q)} - \frac{1}{Q^*}\right)^2, & D(q) \leq Q^*, \\ 0, & D(q) > Q^*, \end{cases}$$

whose gradient is

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

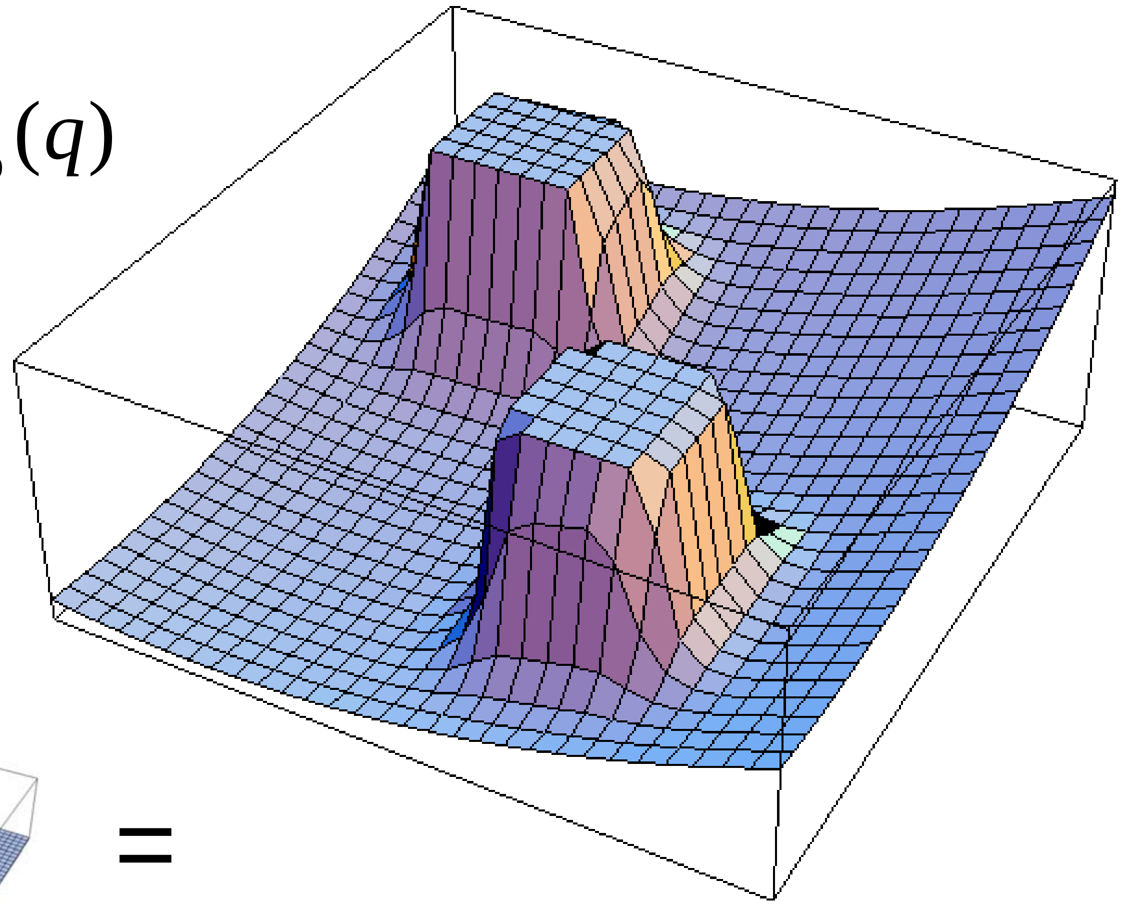
Repulsive Potential

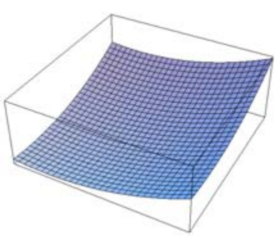


Total Potential Function

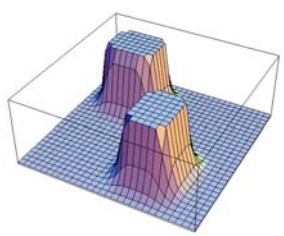
$$U(q) = U_{\text{att}}(q) + U_{\text{rep}}(q)$$

$$F(q) = -\nabla U(q)$$



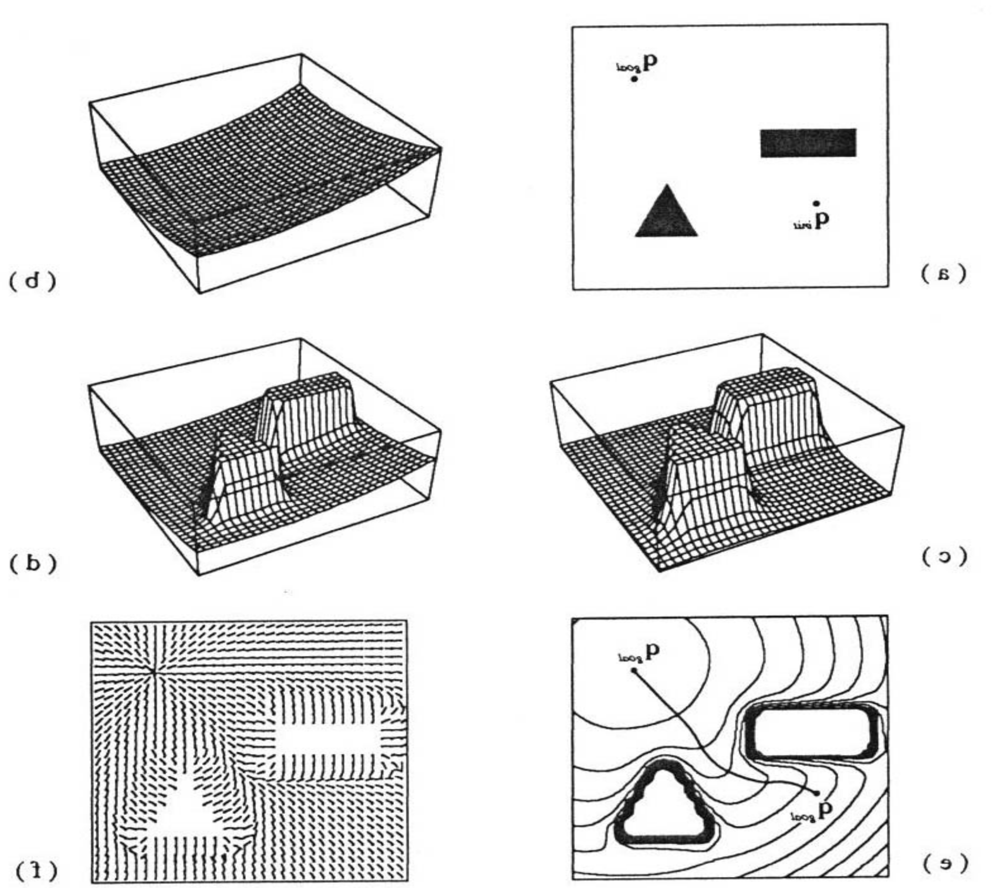


+



=

Potential Fields

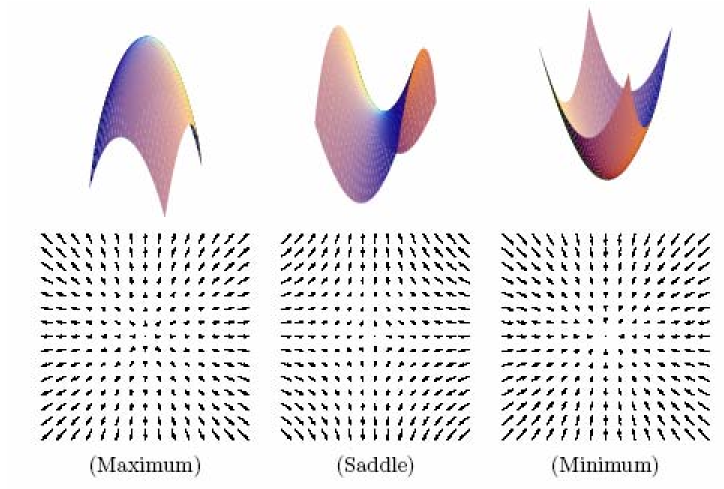


Gradient Descent

- A simple way to get to the bottom of a potential

$$\dot{c}(t) = -\nabla U(c(t)).$$

- A *critical point* is a point x s.t. q^* where $\nabla U(q^*) = 0.$
  - Equation is stationary at a critical point
  - Max, min, saddle
  - Stability?



The Hessian

- For a 1-d function, how do we know we are at a unique minimum (or maximum)?

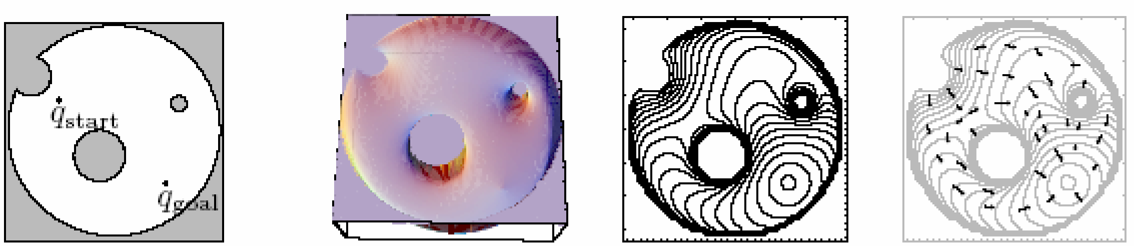
- The Hessian is the $m \times m$ matrix of second derivatives

- If the Hessian is nonsingular ($\mathrm{Det}(H) \neq 0$), the critical point is a unique point
  - if H is positive definite (x^t H x > 0), a minimum
  - if H is negative definite, a maximum
  - if H is indefinite, a saddle point

Gradient Descent

Gradient Descent:

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

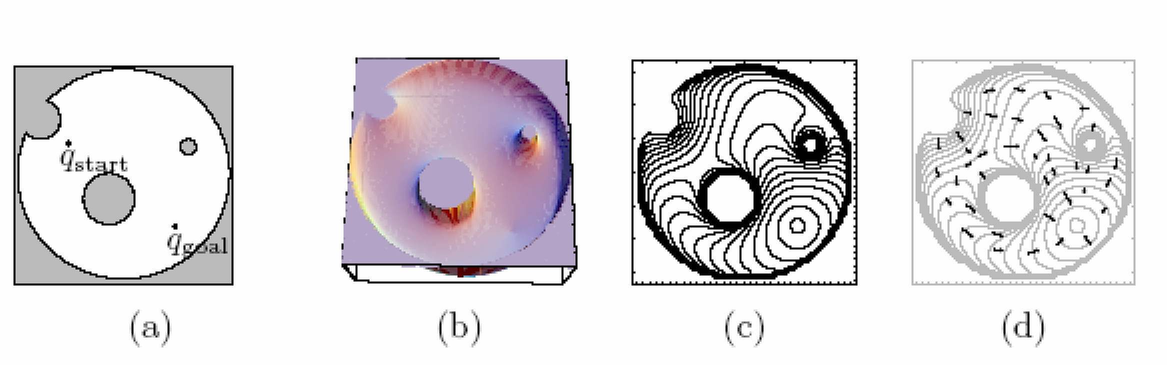


(a) (b) (c) (d)

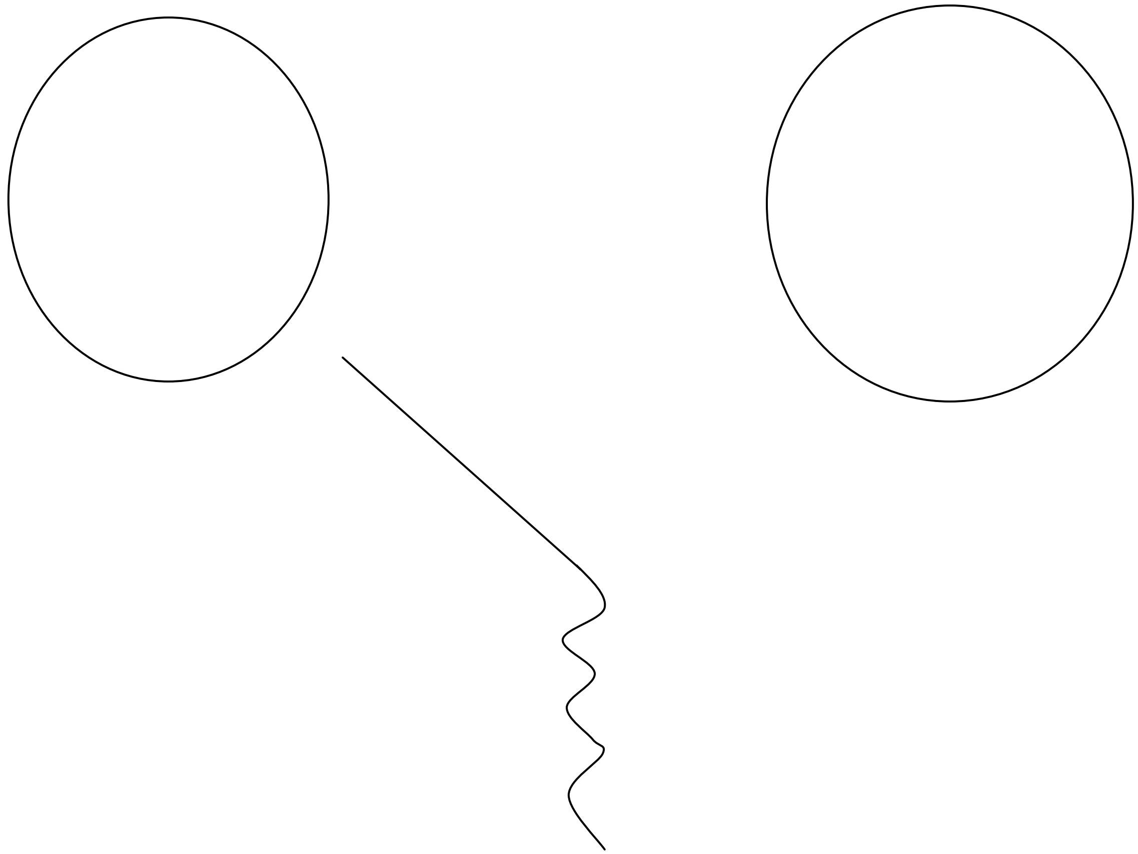
Gradient Descent

Gradient Descent:

- $q(0)=q_{start}$
- $i = 0$
- while $\| \nabla U(q(i)) \| > \varepsilon$ do
  - $q(i+1) = q(i) - \alpha(i) \nabla U(q(i))$
  - $i=i+1$



Numerically “Smoother” Path

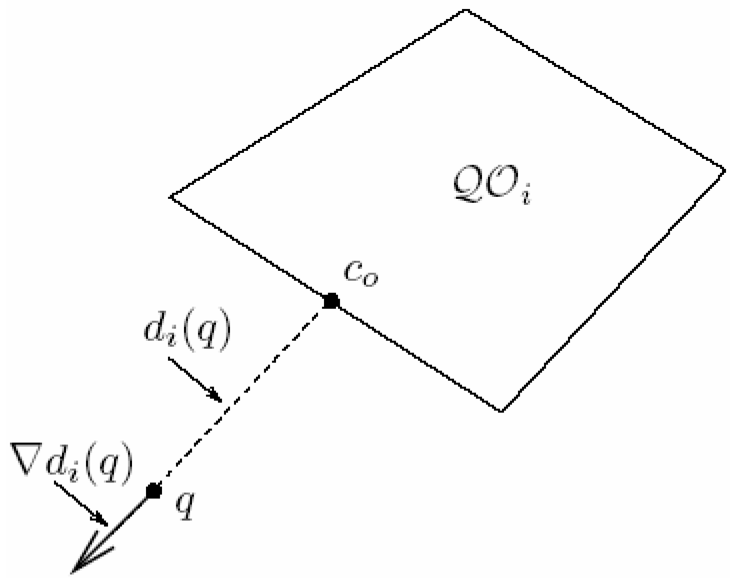


Single Object Distance

$$d_i(q) = \min_{c \in \mathcal{QO}_i} d(q,c)$$

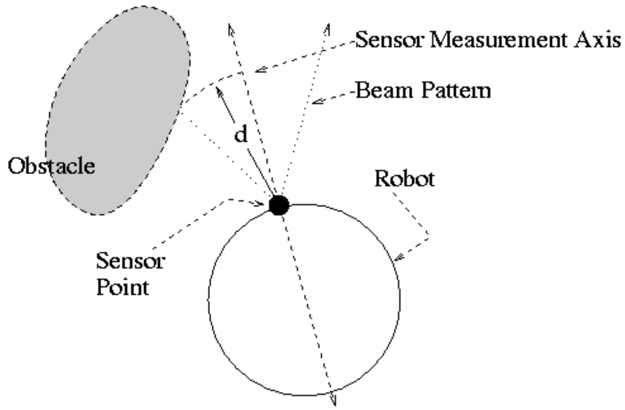
$$\nabla d_i(q) = \frac{q-c}{d(q,c)}$$

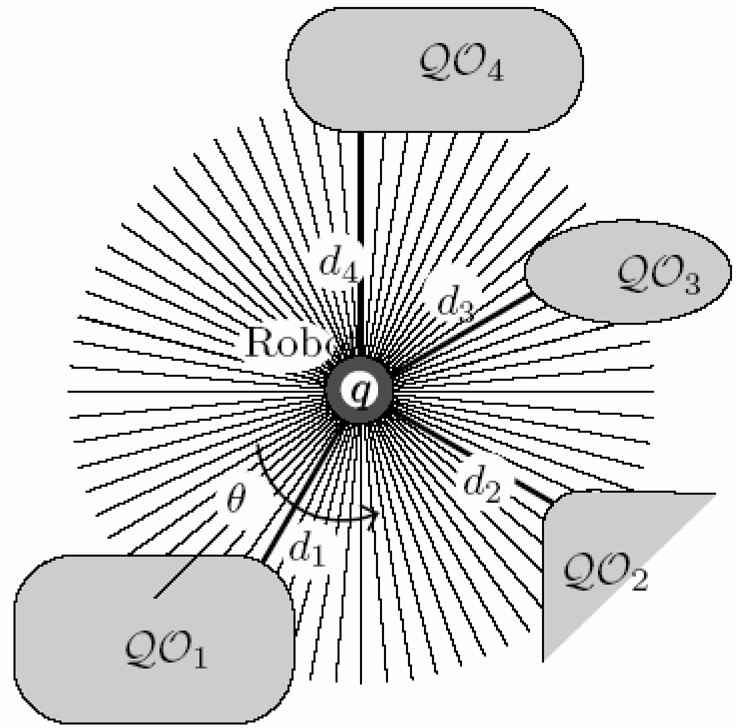
$$U_{\mathrm{rep_i}}(q) = \begin{cases} \frac{1}{2}\eta\left(\frac{1}{d_i(q)} - \frac{1}{Q_i^*}\right)^2, & \text{if } d_i(q) \le Q_i^* \\ 0, & \text{if } d_i(q) > Q_i^* \end{cases}$$



$$U_{\mathrm{rep}}(q) = \sum_{i=1}^{n} U_{\mathrm{rep_i}}(q)$$

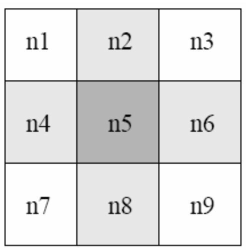
Compute Distance: Sensor Information



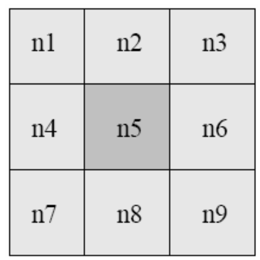


Computing Distance: Use a Grid

- use a discrete version of space and work from there
  - The Brushfire algorithm is one way to do this
    - need to define a grid on space
    - need to define connectivity (4/8)
    - obstacles start with a 1 in grid; free space is zero



4



8

Brushfire Algorithm

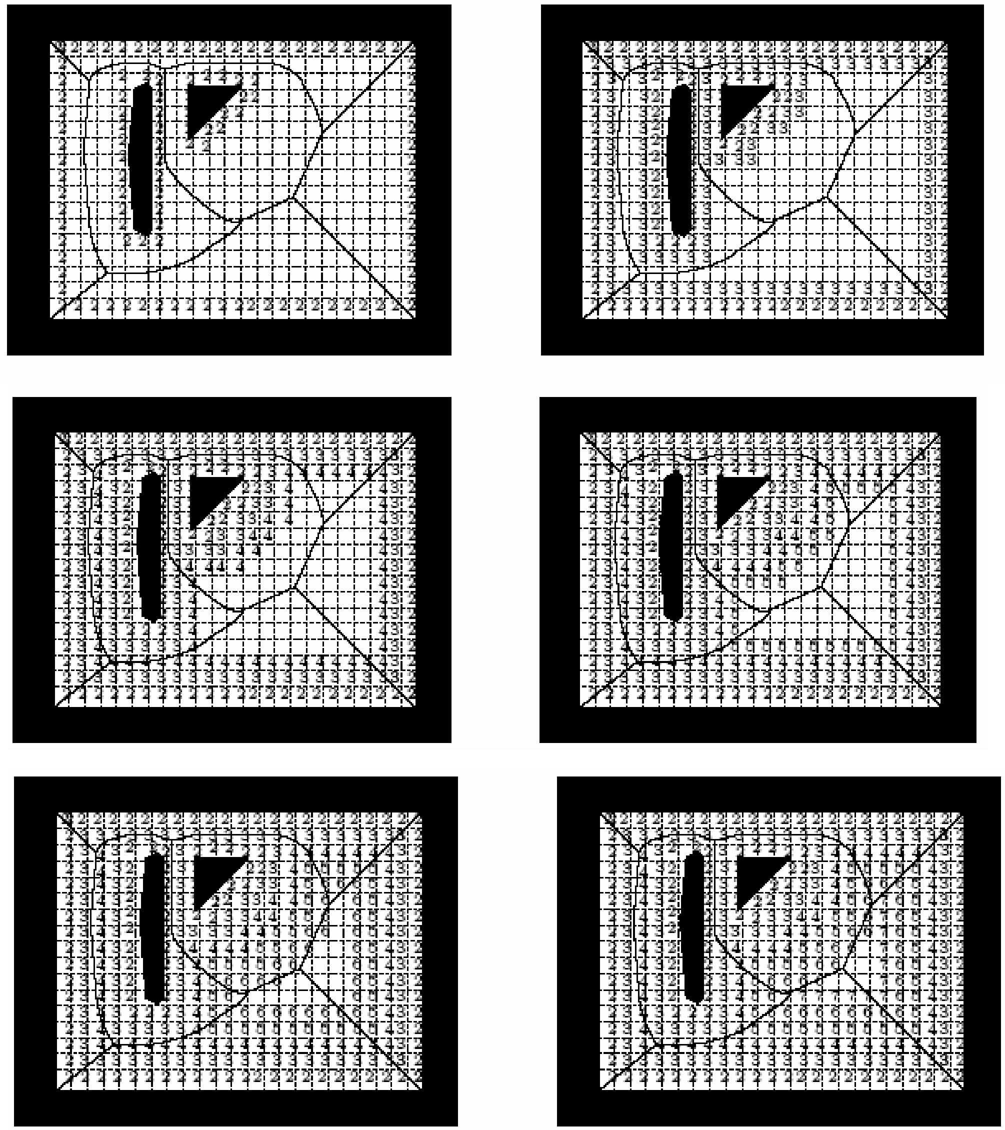
- Initially: create a queue L of pixels on the boundary of all obstacles

- While $L \neq \emptyset$
  - pop the top element t of L
  - if $d(t) = 0$,
    - set $d(t)$ to $1+\min_{t' \in N(t), d(t) \neq 0} d(t')$
    - Add all $t' \in N(t)$ with $d(t)=0$ to L (at the end)

- The result is a distance map d where each cell holds the minimum distance to an obstacle.

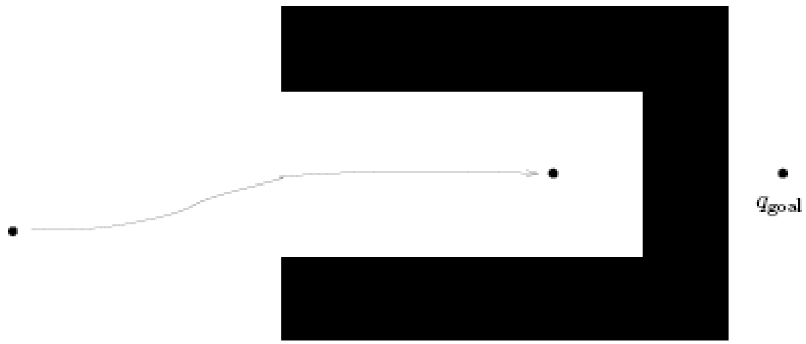
- The gradient of distance is easily found by taking differences with all neighboring cells.

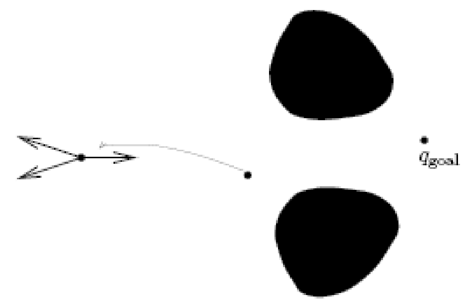
Brushfire example



Potential Functions Question

- How do we know that we have only a single (global) minimum





- We have two choices:
  - not guaranteed to be a global minimum: do something other than gradient descent (what?)
  - make sure only one global minimum (a navigation function, which we'll see later).

The Wave-front Planner

- Apply the brushfire algorithm starting from the goal

- Label the goal pixel 2 and add all zero neighbors to L
  - While $L \neq \emptyset$
    - pop the top element of L, t
    - set d(t) to $1+\min_{t' \in N(t), d(t) > 1} d(t')$
    - Add all $t' \in N(t)$ with d(t)=0 to L (at the end)

- The result is now a distance for every cell
  - gradient descent is again a matter of moving to the neighbor with the lowest distance value

The 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

The Wavefront in Action (Part 1)

- Starting with the goal, set all adjacent cells with “0” to the current cell + 1
  - 4-Point Connectivity or 8-Point Connectivity?
  - Your Choice. We’ll use 8-Point Connectivity in our example

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

The Wavefront in Action (Part 2)

- Now repeat with the modified cells
  - This will be repeated until no 0’s are adjacent to cells with values >= 2
    - 0’s will only remain when regions are unreachable

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

The Wavefront in Action (Part 3)

- Repeat again...

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

The Wavefront in Action (Part 4)

- And again...

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

The Wavefront in Action (Part 5)

- And again until...

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

The Wavefront in Action (Done)

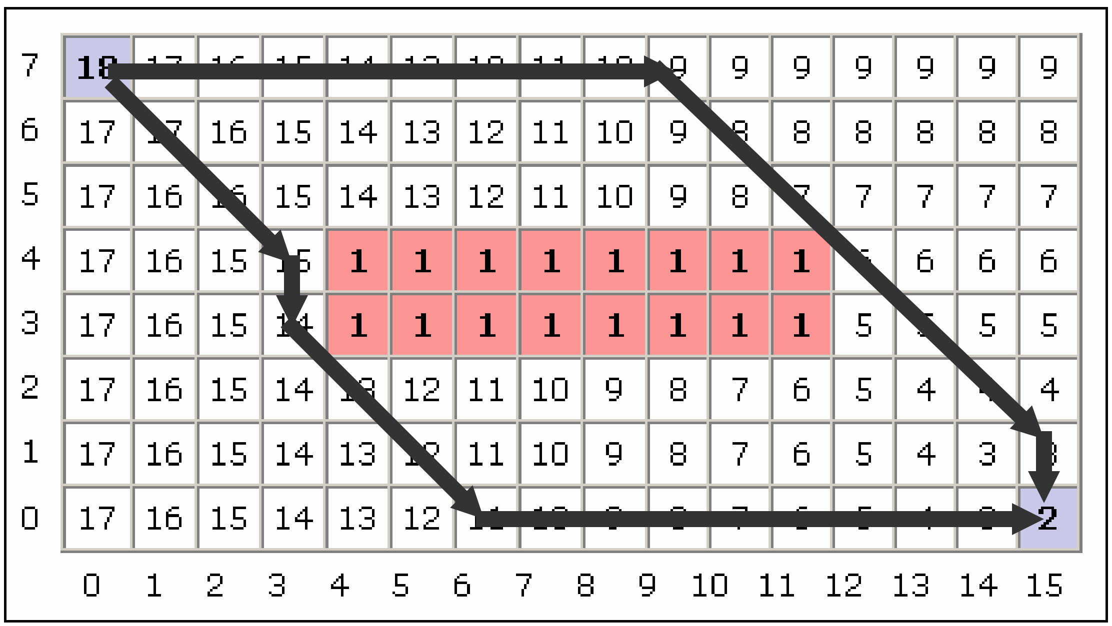
- You're done
  - Remember, 0's should only remain if unreachable regions exist

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

The 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 shown



Wavefront (Overview)

- Divide the space into a grid.
- Number the squares starting at the start in either 4 or 8 point connectivity starting at the goal, increasing till you reach the start.
- Your path is defined by any uninterrupted sequence of decreasing numbers that lead to the goal.