Fast Computation of Robot-Obstacle Interactions in Nonholonomic Trajectory Deformation

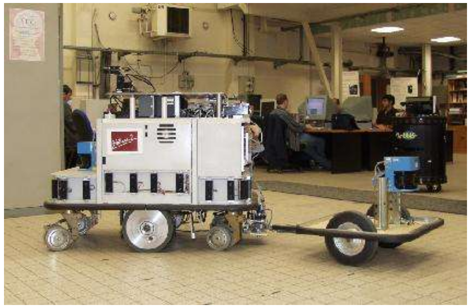
Olivier Lefebvre, Florent Lamiraux, David Bonnafous

LAAS-CNRS, Toulouse, France

April, 18-22th 2005 ICRA 2005 Barcelona, Spain

Motivation and context

- Navigation in cluttered environment for nonholonomic systems

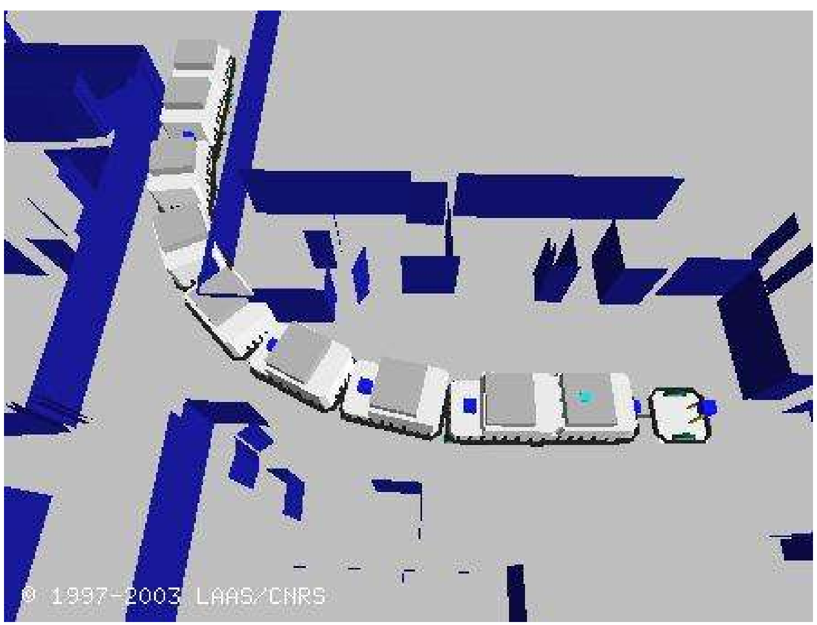


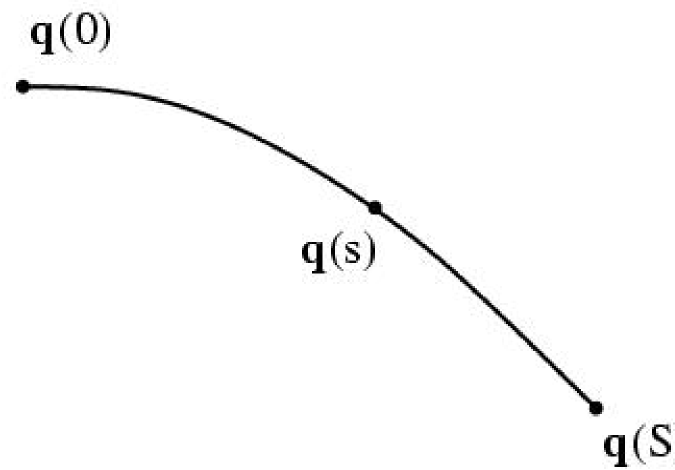
- Reactive obstacle avoidance
  - poor localization and map imprecision
  - unexpected obstacles

Nonholonomic trajectory deformation

Deform the trajectory in order to:

- move away from obstacles
- keep nonholonomic constraints satisfied

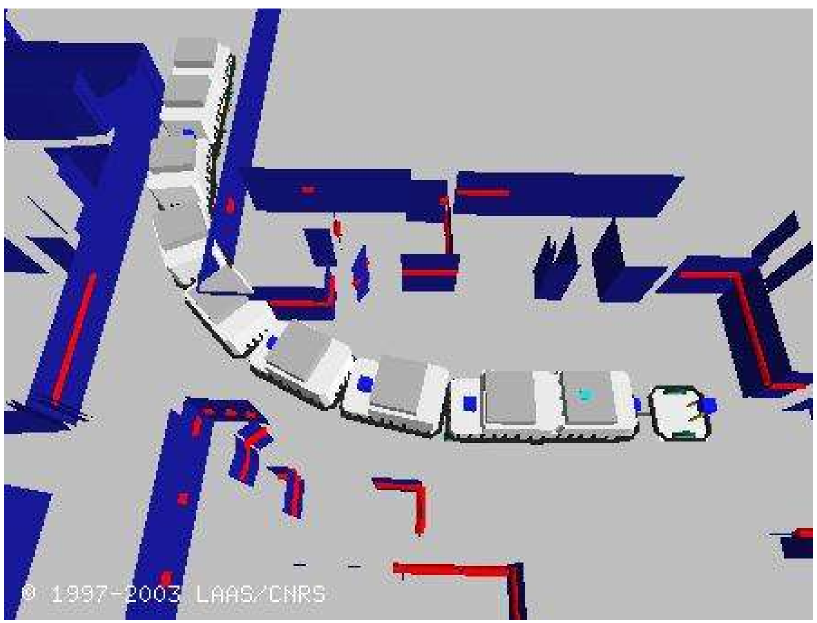


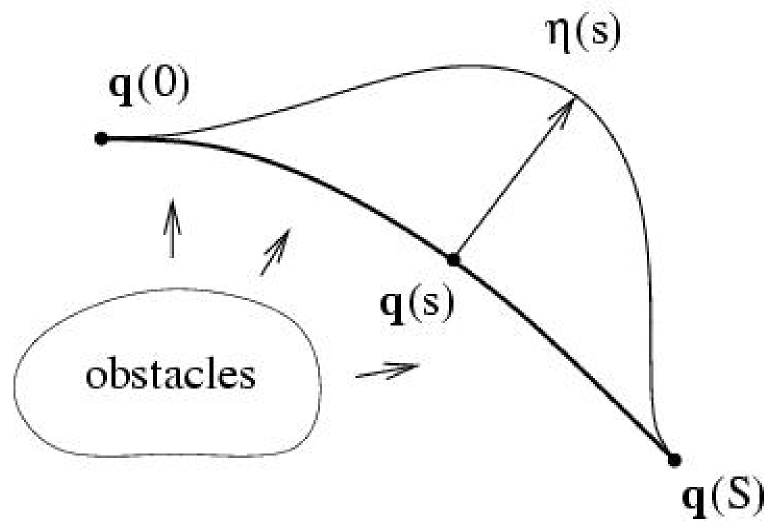


Nonholonomic trajectory deformation

Deform the trajectory in order to:

- move away from obstacles
- keep nonholonomic constraints satisfied

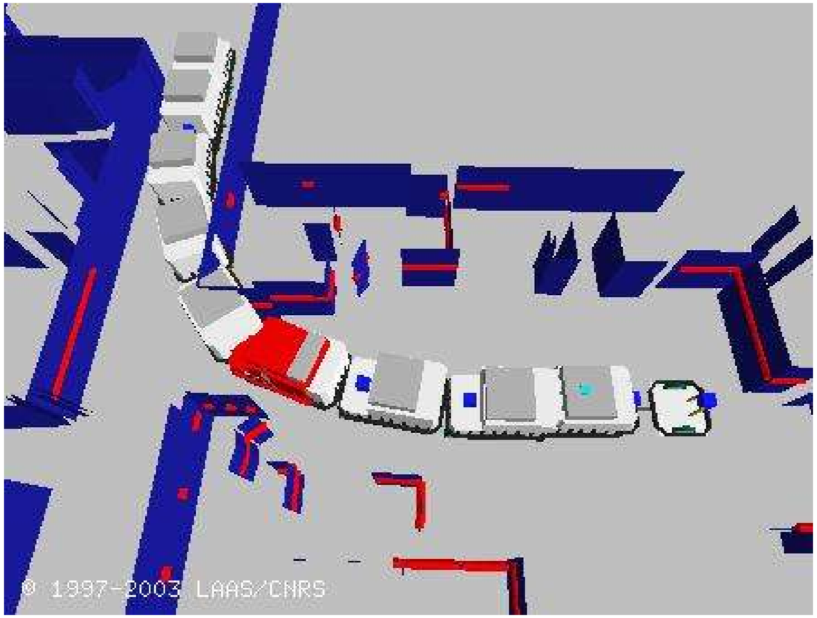


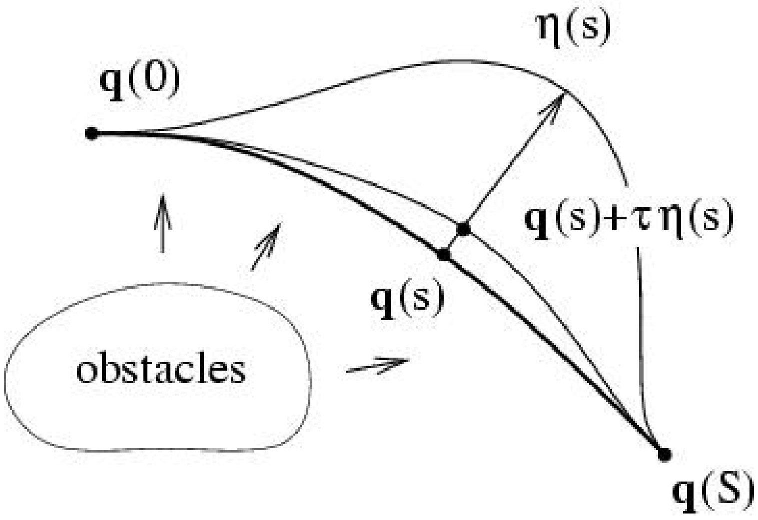


Nonholonomic trajectory deformation

Deform the trajectory in order to:

- move away from obstacles
- keep nonholonomic constraints satisfied

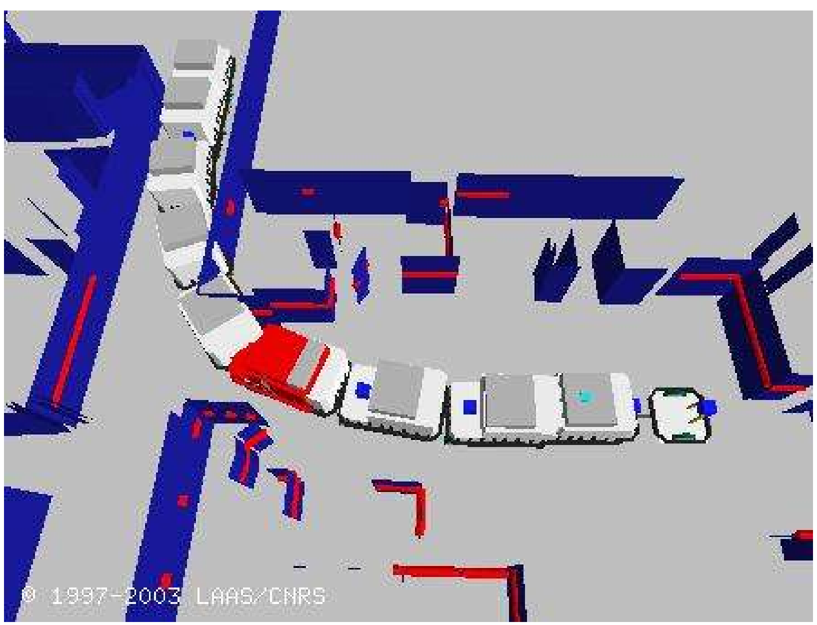


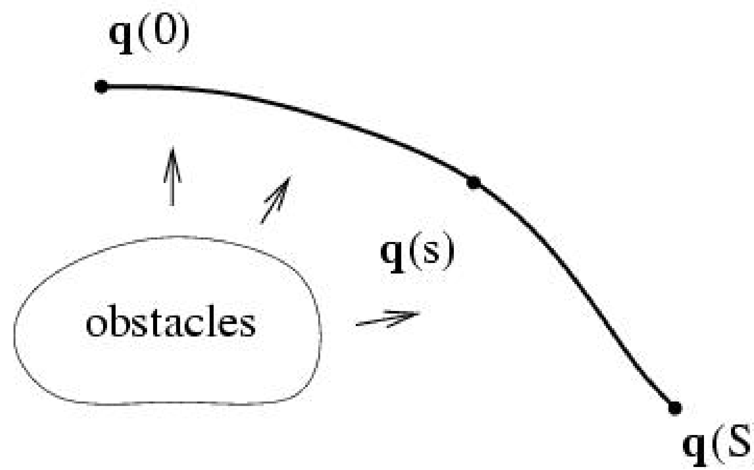


Nonholonomic trajectory deformation

Deform the trajectory in order to:

- move away from obstacles
- keep nonholonomic constraints satisfied

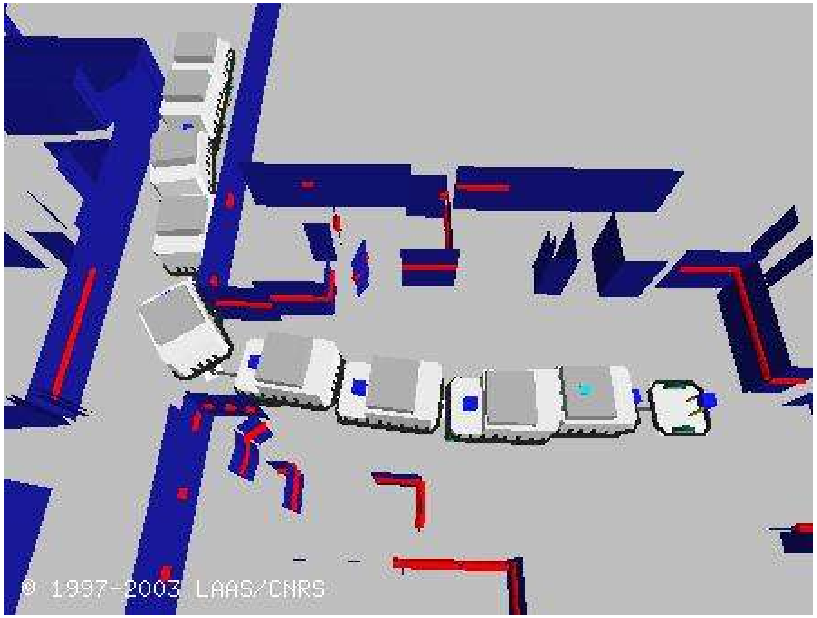


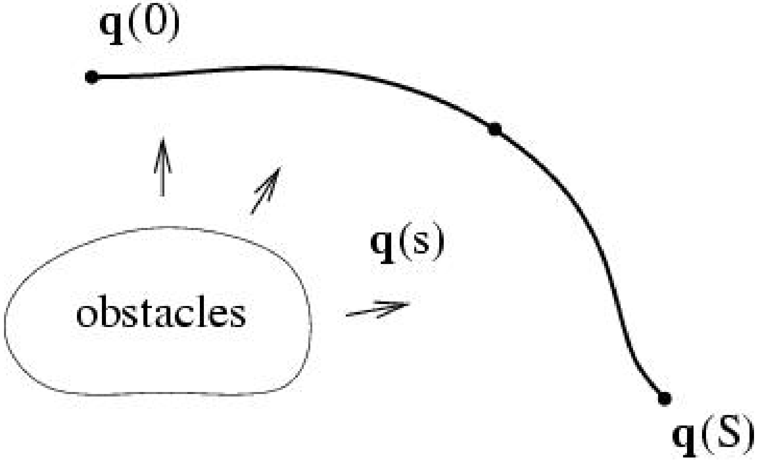


Nonholonomic trajectory deformation

Deform the trajectory in order to:

- move away from obstacles
- keep nonholonomic constraints satisfied





Direction of deformation

How to compute a direction of deformation that moves away from obstacles ?

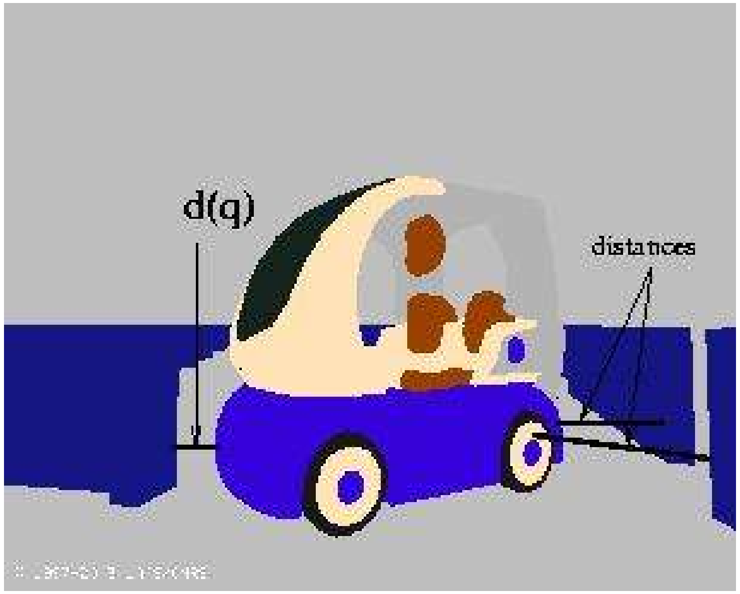
Direction of deformation

How to compute a direction of deformation that moves away from obstacles ?

Artificial potential fields over configuration space *[Khatib 86]*

$$U(\mathbf{q}(s)) = \frac{1}{d(\mathbf{q}(s))}$$

configurations close to obstacles have a higher potential



Direction of deformation

- Potential of a trajectory

$$V(\mathbf{q}) = \int_0^S U(\mathbf{q}(s))ds$$

move away from obstacles <=> make the potential decrease

Direction of deformation

- Potential of a trajectory

$$V(\mathbf{q}) = \int_0^S U(\mathbf{q}(s))ds$$

move away from obstacles <=> make the potential decrease

- Gradient of the potential

$$dV = \int_0^S \frac{\partial U}{\partial \mathbf{q}}(\mathbf{q}(s))\eta(s)ds$$

compute $\eta(s)$ such that dV it is negative

Contribution

$$dV = \int_0^S \frac{\partial U}{\partial \mathbf{q}}(\mathbf{q}(s))\eta(s)ds$$

Contribution

$$dV = \int_0^S \frac{\partial U}{\partial \mathbf{q}}(\mathbf{q}(s))\eta(s)ds$$

- compute gradient of potential over configuration space without closed-form expression

Contribution

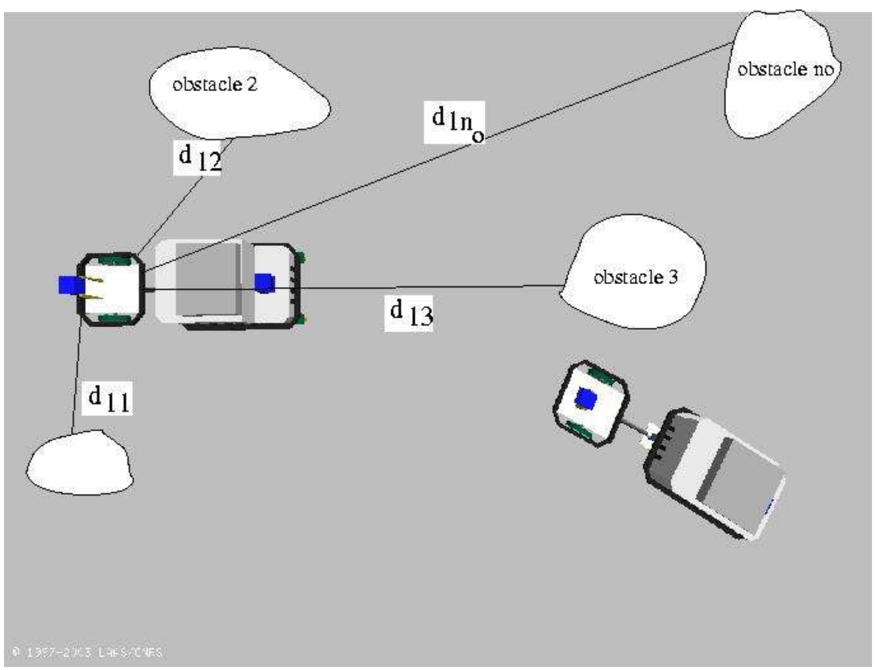
$$dV = \int_0^S \frac{\partial U}{\partial \mathbf{q}}(\mathbf{q}(s))\eta(s)ds$$

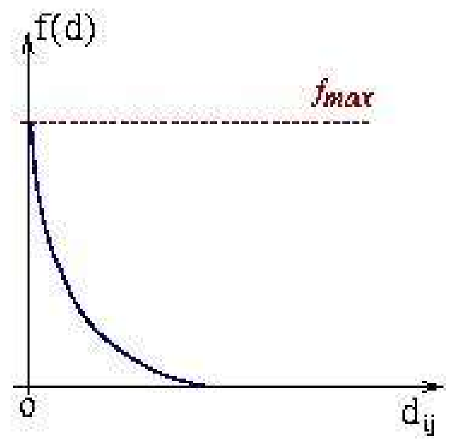
- compute gradient of potential over configuration space without closed-form expression
- optimize computation over the trajectory using spacial coherence

Gradient of Potential

- $U(\mathbf{q})$ is a function of distance to obstacles:

$$U(\mathbf{q}) = \sum_{i=1}^{n_b} \sum_{j=1}^{n_o} f(d_{ij}(\mathbf{q}))$$





Gradient of Potential

- We need the gradient of the potential of a configuration

$$\frac{\partial U}{\partial \mathbf{q}}(\mathbf{q}) = \sum_{i=1}^{n_b}\sum_{j=1}^{n_o} f'(d_{ij}(\mathbf{q}))\frac{\partial d_{ij}}{\partial \mathbf{q}}(\mathbf{q})$$

known

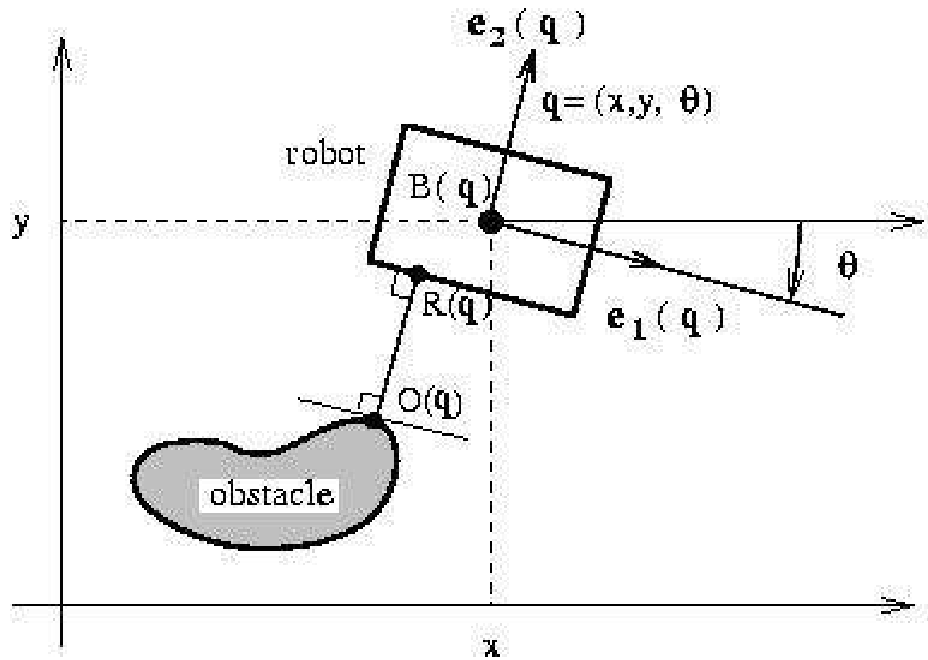
known

how to
compute it ?

Gradient of Potential

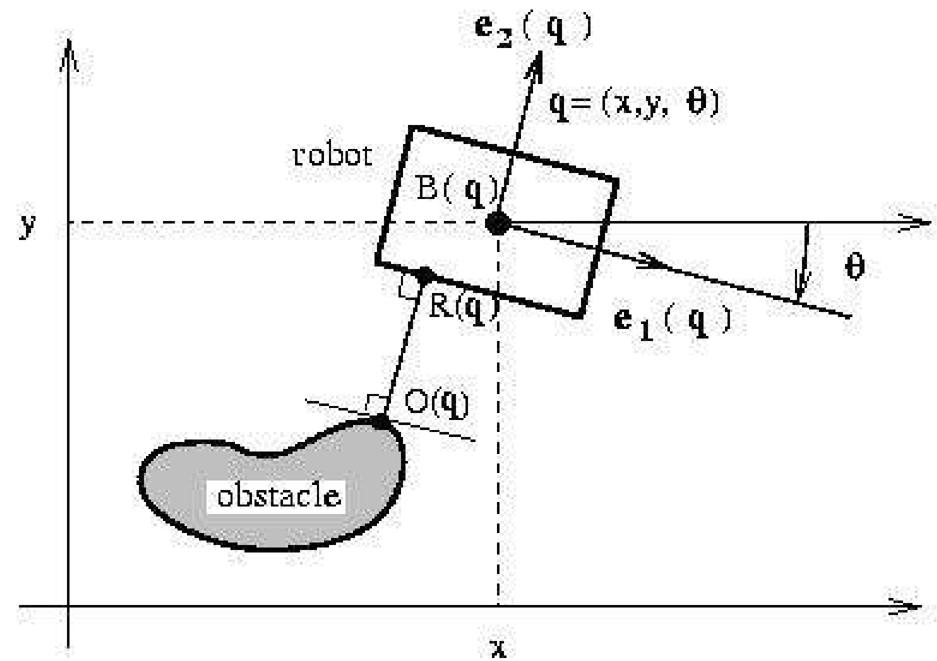
- derivative of distance:

$$\frac{\partial d_{ij}}{\partial \mathbf{q}}(\mathbf{q}) = \frac{(\mathbf{O}(\mathbf{q}) - \mathbf{R}(\mathbf{q}))^T}{\|\mathbf{O}(\mathbf{q}) - \mathbf{R}(\mathbf{q})\|} \left(\frac{\partial \mathbf{O}}{\partial \mathbf{q}}(\mathbf{q}) - \frac{\partial \mathbf{R}}{\partial \mathbf{q}}(\mathbf{q}) \right)$$



Composition of motions

$$\mathbf{R}(\mathbf{q}) = \sum_{l=1}^{d} \rho_l(\mathbf{q})\mathbf{e}_l(\mathbf{q})$$



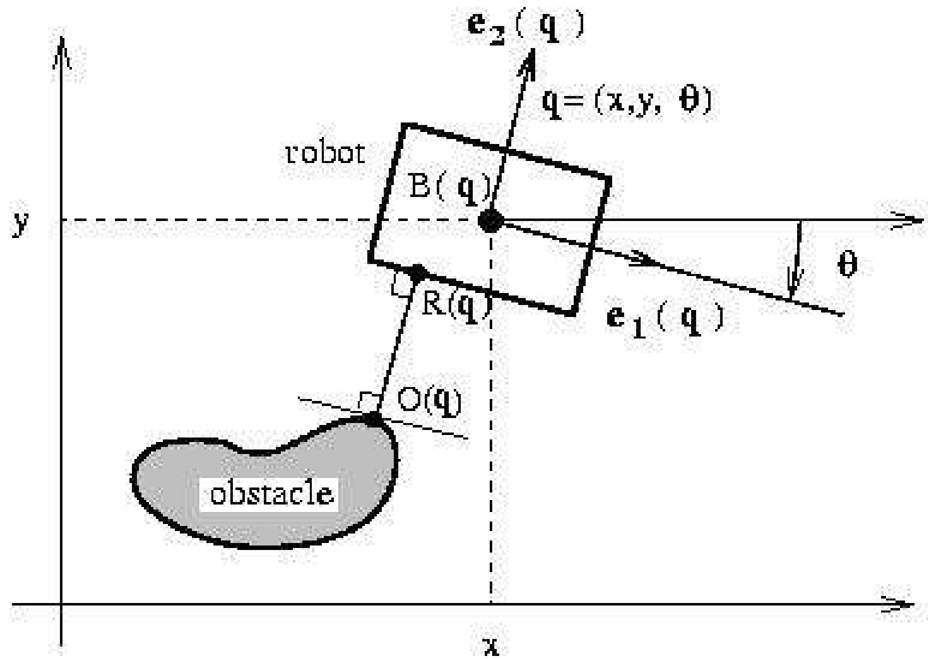
$$\frac{\partial \mathbf{R}}{\partial \mathbf{q}}(\mathbf{q}) = \sum_{l=1}^{d} \frac{\partial \rho_l}{\partial \mathbf{q}}(\mathbf{q})\mathbf{e}_l(\mathbf{q}) +$$ ← motion of R(q) on the robot

$$\sum_{l=1}^{d} \rho_l(\mathbf{q}) \frac{\partial \mathbf{e}_l(\mathbf{q})}{\partial \mathbf{q}}$$ ← absolute motion of coincinding point

Composition of motions

- motion of R(q) is orthogonal to

$$(\mathbf{O}(\mathbf{q}) - \mathbf{R}(\mathbf{q}))^T$$



$$\frac{\partial \mathbf{R}}{\partial \mathbf{q}}(\mathbf{q}) = \sum_{l=1}^{d} \frac{\partial \rho_l}{\partial \mathbf{q}}(\mathbf{q})\mathbf{e}_l(\mathbf{q}) +$$

← motion of R(q) on the robot

$$\sum_{l=1}^{d} \rho_l(\mathbf{q}) \frac{\partial \mathbf{e}_l(\mathbf{q})}{\partial \mathbf{q}}$$

← absolute motion of coinciding point

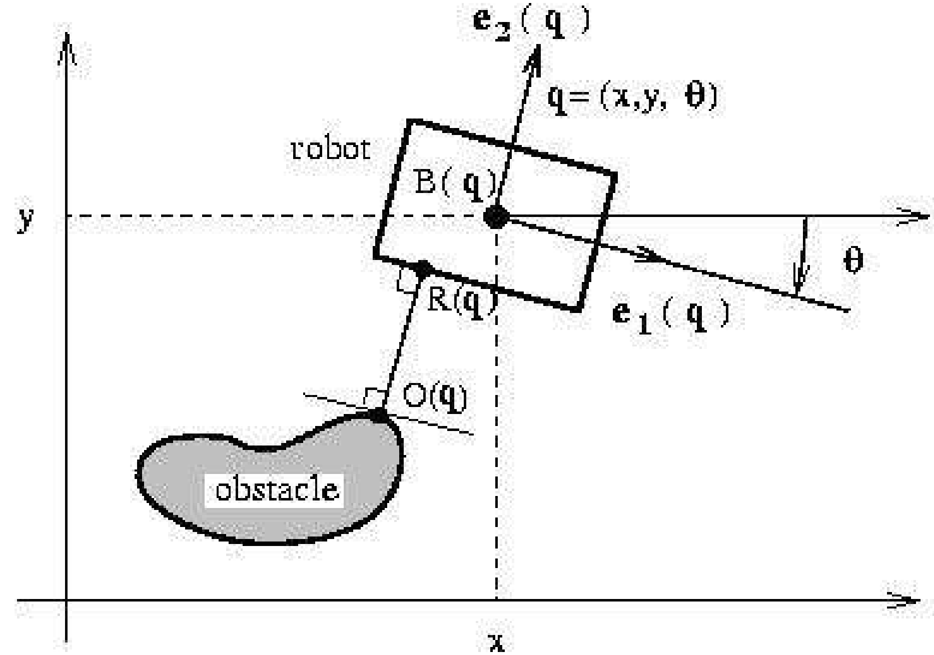
Composition of motions

- motion of R(q) is orthogonal to:

$$(\mathbf{O}(\mathbf{q}) - \mathbf{R}(\mathbf{q}))^T$$

and we need to compute:

$$\frac{\partial d_{ij}}{\partial \mathbf{q}}(\mathbf{q}) = \frac{(\mathbf{O}(\mathbf{q}) - \mathbf{R}(\mathbf{q}))^T}{\|\mathbf{O}(\mathbf{q}) - \mathbf{R}(\mathbf{q})\|} \left(\frac{\partial \mathbf{O}}{\partial \mathbf{q}}(\mathbf{q}) - \frac{\partial \mathbf{R}}{\partial \mathbf{q}}(\mathbf{q}) \right)$$



$$\frac{\partial \mathbf{R}}{\partial \mathbf{q}}(\mathbf{q}) = \sum_{l=1}^{d} \frac{\partial \rho_l}{\partial \mathbf{q}}(\mathbf{q}) \mathbf{e}_l(\mathbf{q}) +$$

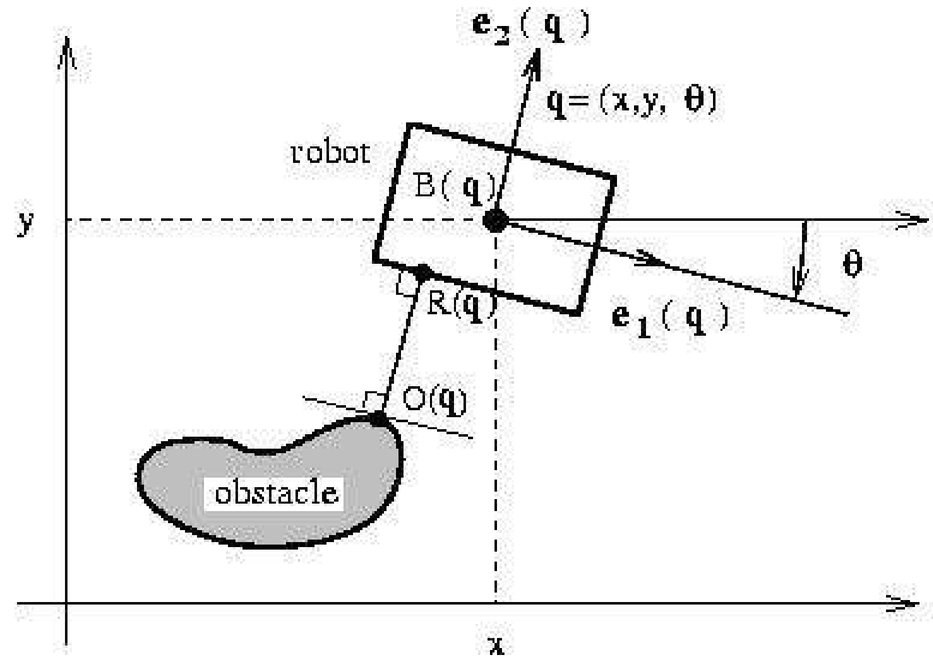
← ~~motion of R(q) on the robot~~

$$\sum_{l=1}^{d} \rho_l(\mathbf{q}) \frac{\partial \mathbf{e}_l(\mathbf{q})}{\partial \mathbf{q}}$$

← absolute motion of coinciding point

Composition of motions

- same reasoning with obstacles
- fixed obstacles



$$\frac{\partial \mathbf{O}}{\partial \mathbf{q}}(\mathbf{q}) = \sum_{l=1}^{d} \frac{\partial \rho_l}{\partial \mathbf{q}}(\mathbf{q})\mathbf{e}_l(\mathbf{q}) +$$

← ~~motion of O(q) on the obstacle~~

$$\sum_{l=1}^{d} \rho_l(\mathbf{q}) \frac{\partial \mathbf{e}_l(\mathbf{q})}{\partial \mathbf{q}}$$

← ~~absolute motion of coinciding point~~

Gradient of configuration space potential field

- No closed-form required if it is a function of distance to obstacles
- compute $\frac{\partial \mathbf{R}_{\in body}}{\partial \mathbf{q}}$ only

Filter useless Robot-Obstacles Interaction pairs

$$dV = \int_0^S \frac{\partial U}{\partial \mathbf{q}}(\mathbf{q}(s))\eta(s)ds$$

Filter useless Robot-Obstacles Interaction pairs

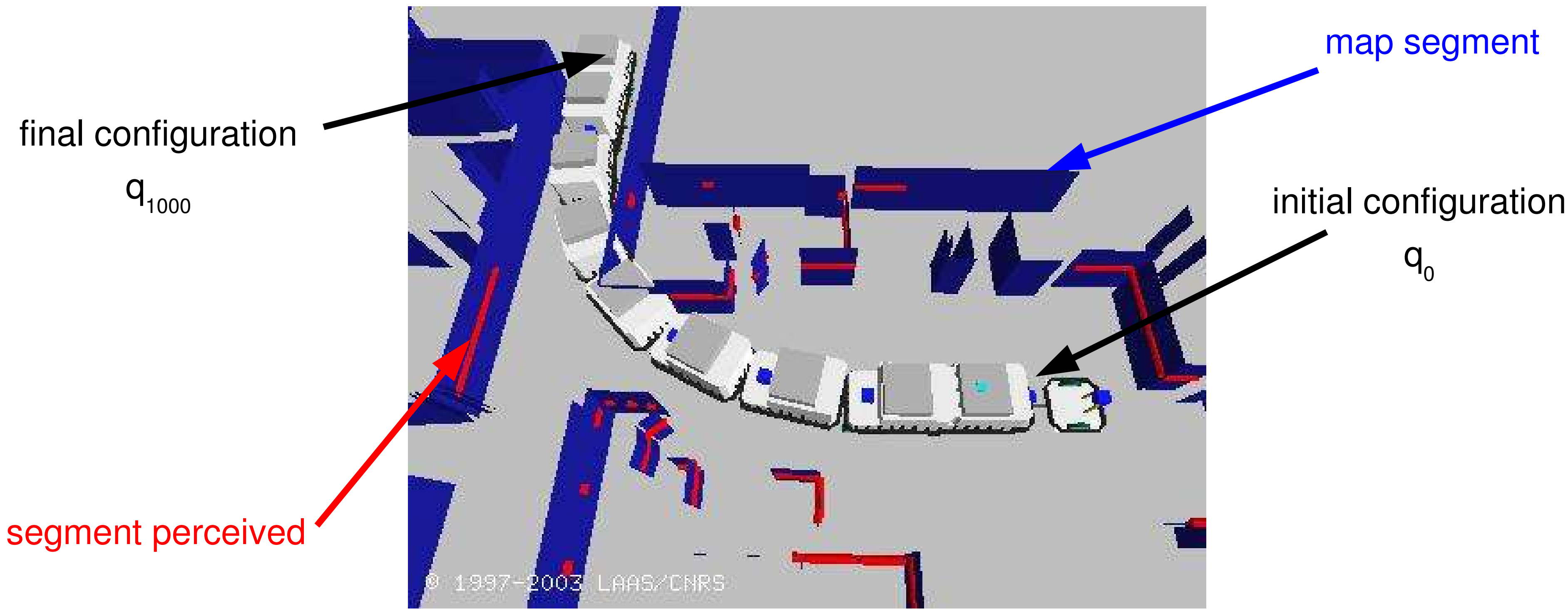
$$dV = \int_0^S \frac{\partial U}{\partial \mathbf{q}}(\mathbf{q}(s))\eta(s)ds$$

Computations are done for all configurations along the planned trajectory

How to reduce the complexity ?

Filter useless Robot-Obstacles Interaction pairs

- Given a sensor perception and a discretized trajectory

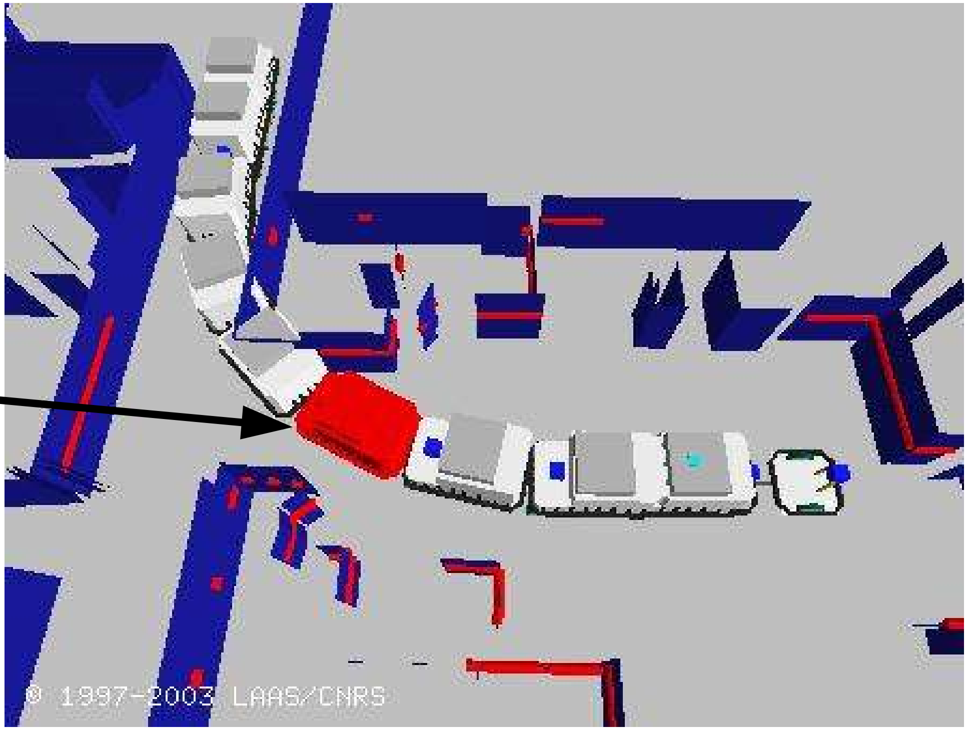


Filter useless Robot-Obstacles Interaction pairs

1. compute index of collision *(n_{conf} * $n_{obstacles}$)*

each configuration is tested against collision with each obstacle

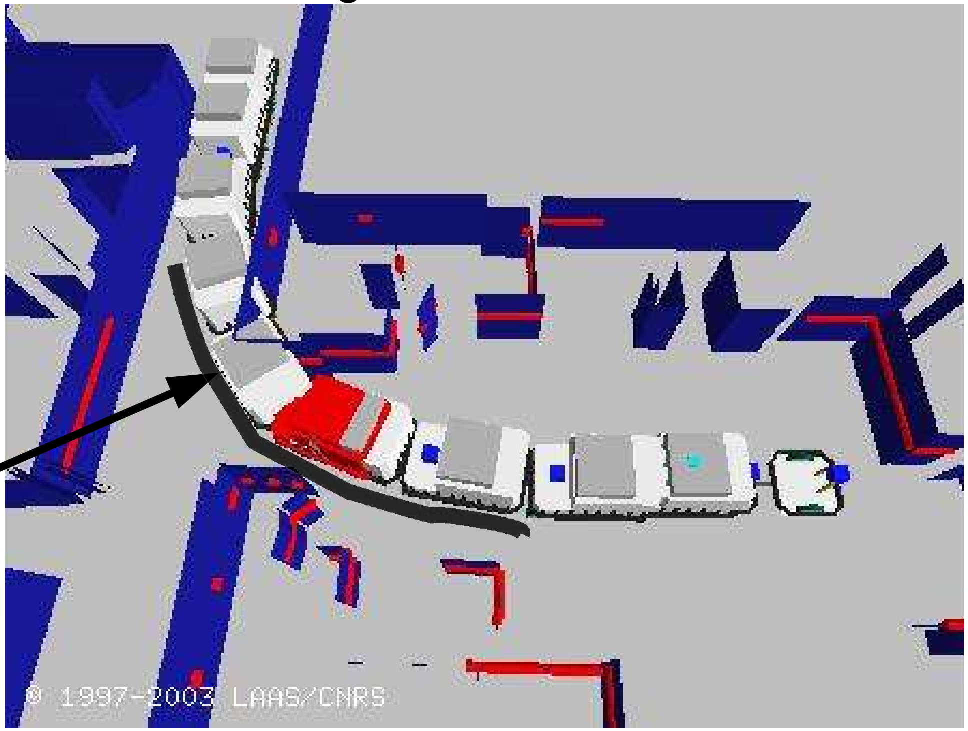
first configuration
in collision



Filter useless Robot-Obstacles Interaction pairs

2. compute the gradient of the potential field *($n_{interval}$ * $n_{obstacles}$)*

gradient of potential created by every obstacle is computed for each configuration along the deformation interval

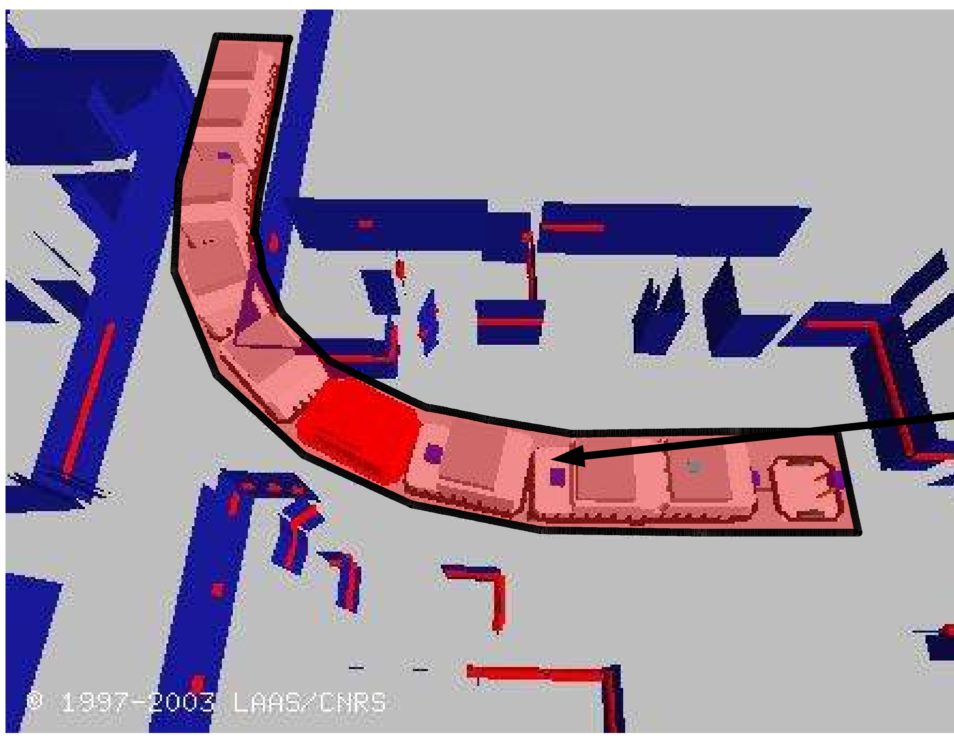


deformation interval
($n_{interval}$ configurations)

Useless Robot-Obstacle pairs

- Most of the robot-obstacle pairs are useless:

collision : minimal clearance distance ρ



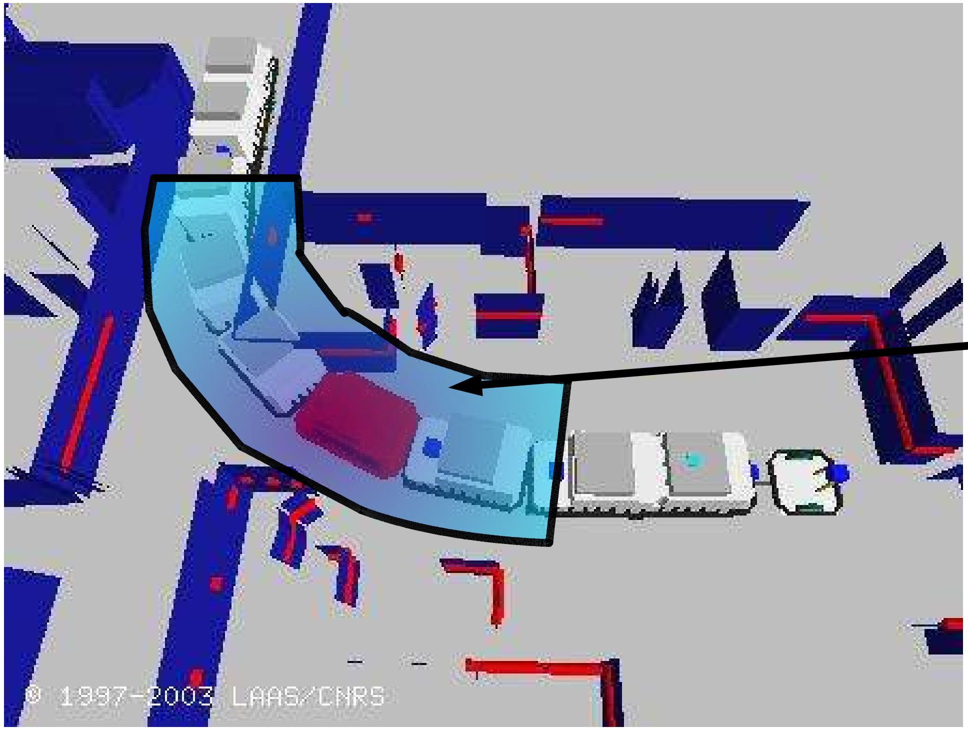
collision tunnel for body trailer

swept volume = $w_{trailer} + 2\rho$

Useless Robot-Obstacle pairs

- Most of the robot-obstacle pairs are useless:

gradient of the potential field : maximal distance of influence ρ



influence distance tunnel

swept volume = $w_{trailer} + 2\rho$

Filter using spatial coherence

- spatial coherence:
  - obstacles “far” from q_s are still “far” from q_{s+1}

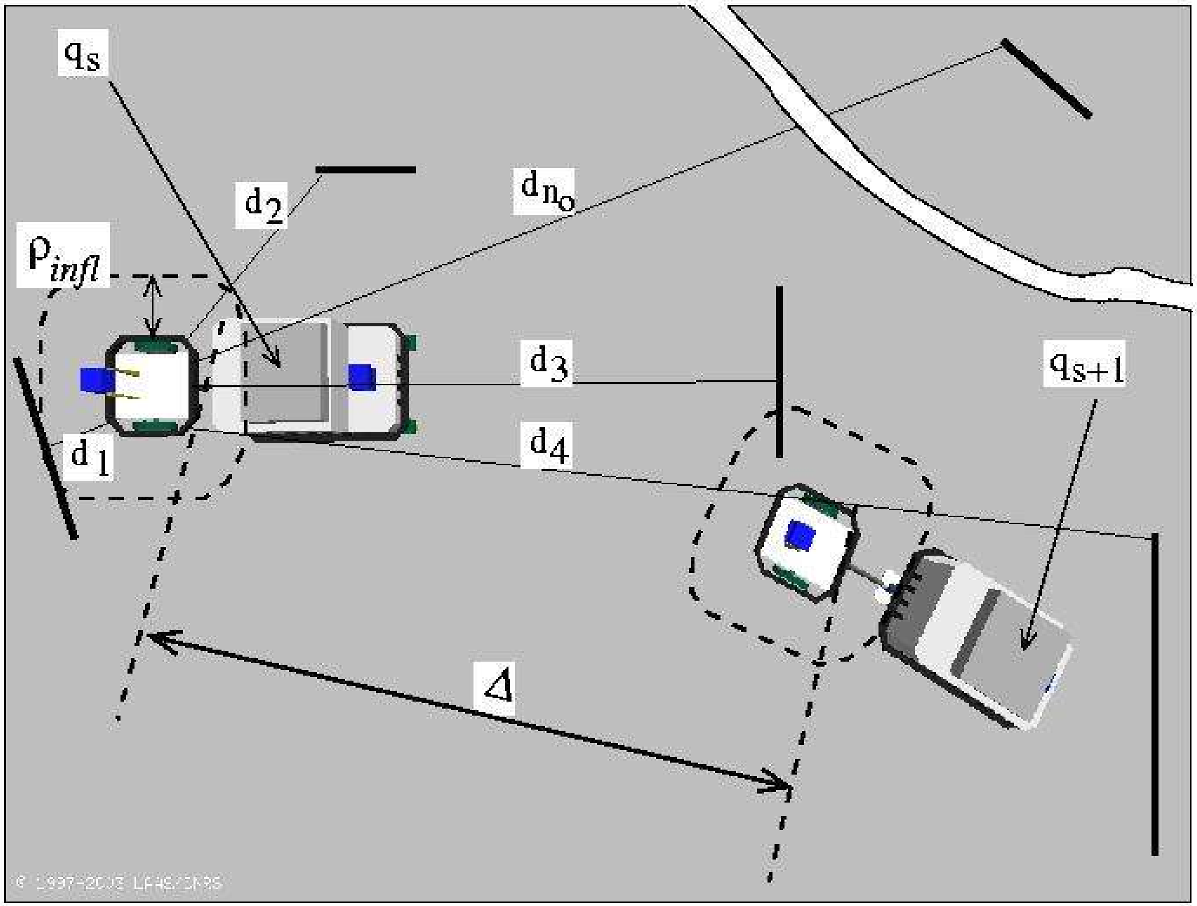
maximal distance traveled by points of body B_i between q_s and q_{s+1} : Δ

distance between 2 compact subsets : $d(\mathcal{A},\mathcal{B}) = \min_{a\in\mathcal{A},b\in\mathcal{B}} \|b-a\|$

Filter using spatial coherence

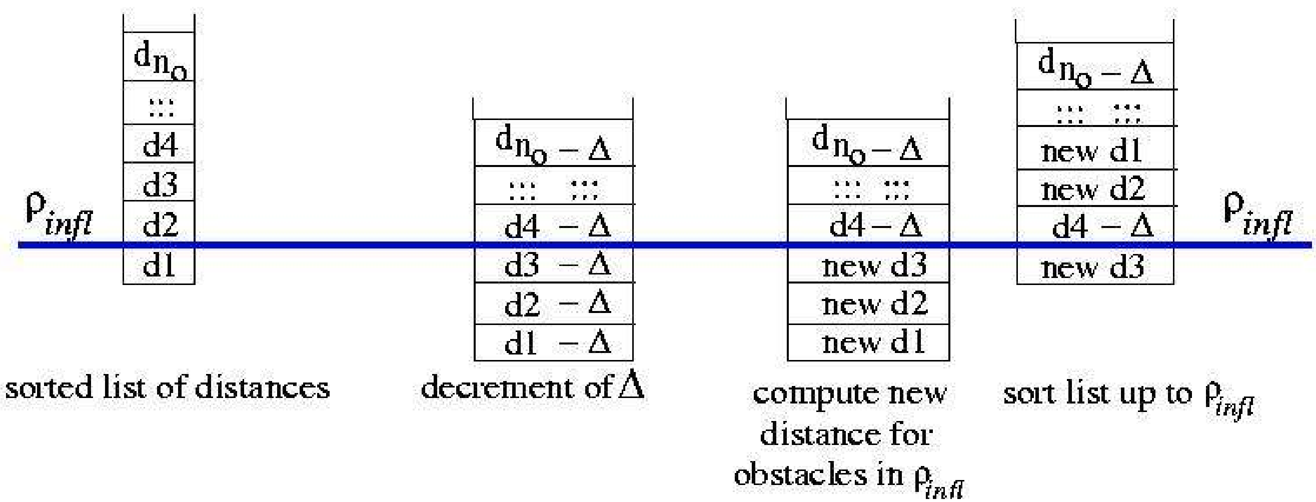
- spatial coherence: example with the trailer

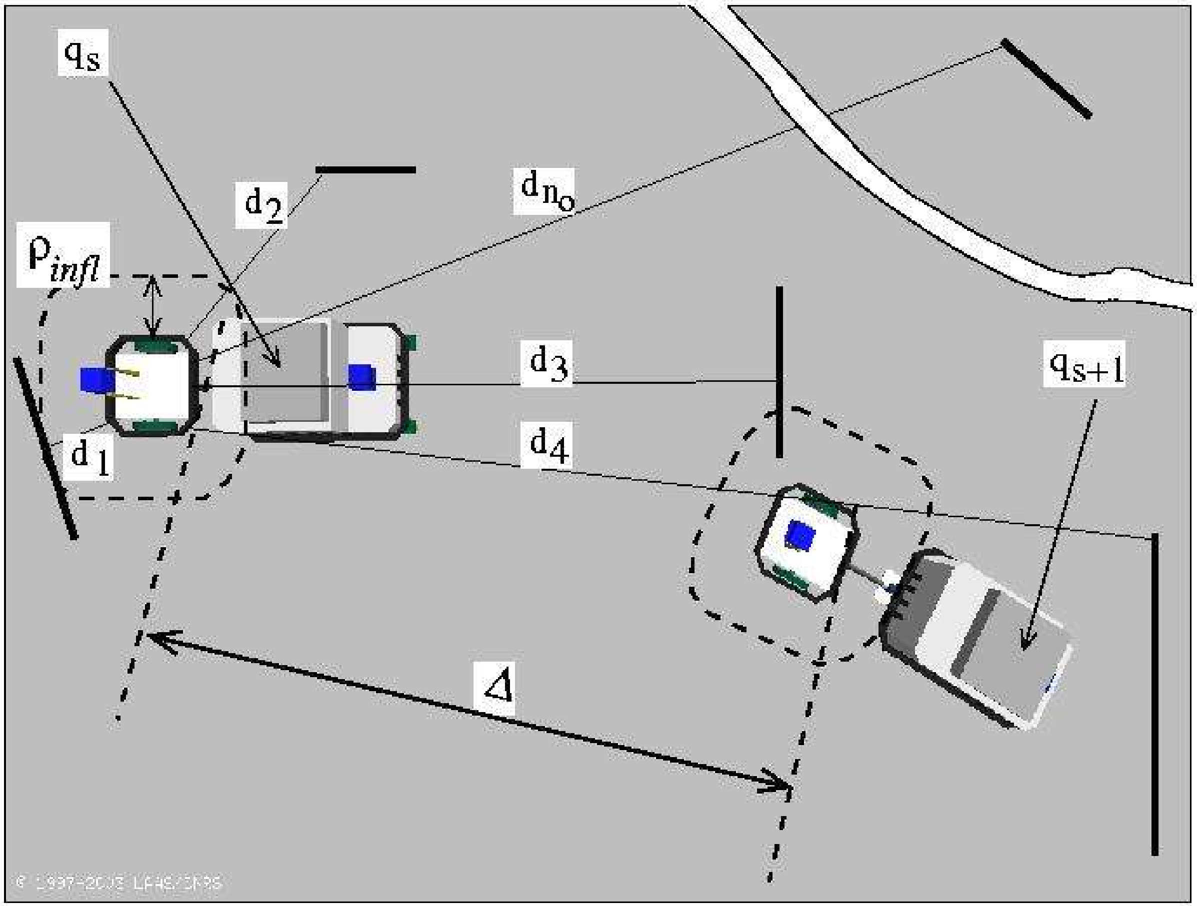
$$d(\mathcal{B}_i(\mathbf{q}_1), \mathcal{O}) \geq d_1 \quad \Rightarrow \quad d(\mathcal{B}_i(\mathbf{q}_2), \mathcal{O}) \geq \max(d_1 - \Delta, 0)$$



- compute a sorted list of distances to obstacles

distance < ρ_{infl} => usefull robot-obstacle pair





- compute a sorted list of distances to obstacles

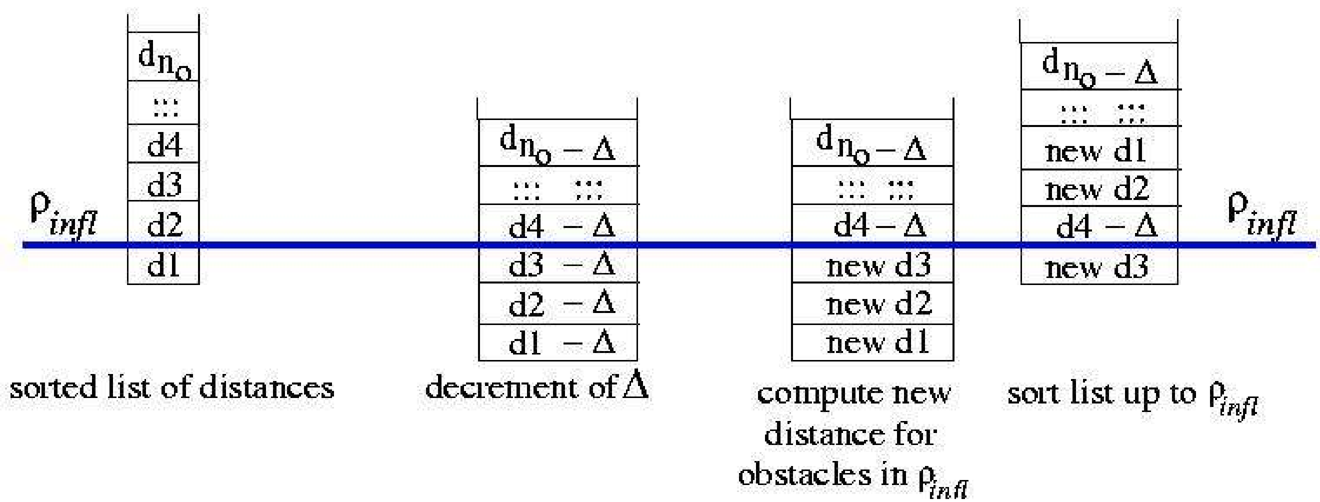
distance < ρ_{infl} => usefull robot-obstacle pair

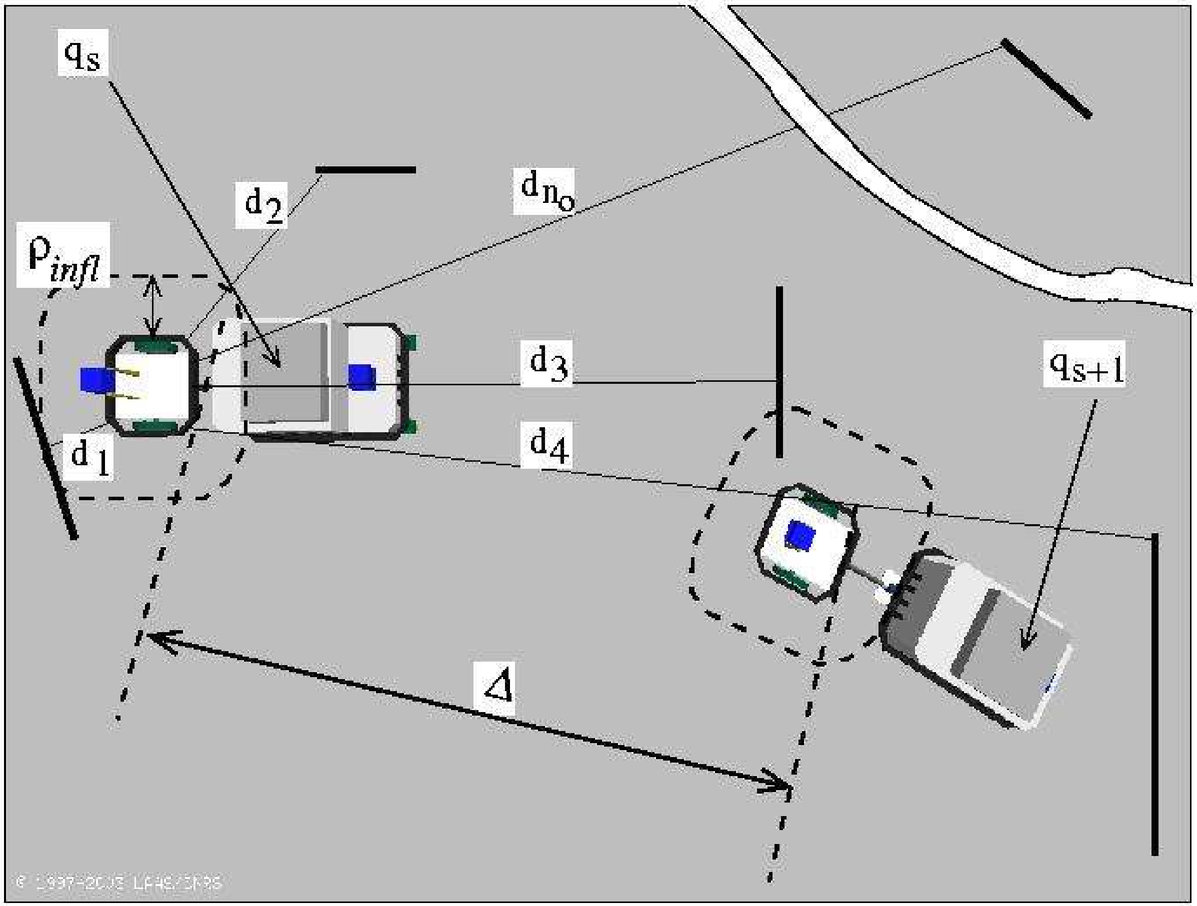
- between two configurations

subtract Δ to all elements

distance < ρ => recompute exact distance

insert it in the list





- compute a sorted list of distances to obstacles

  distance < ρ_{infl} => usefull robot-obstacle pair

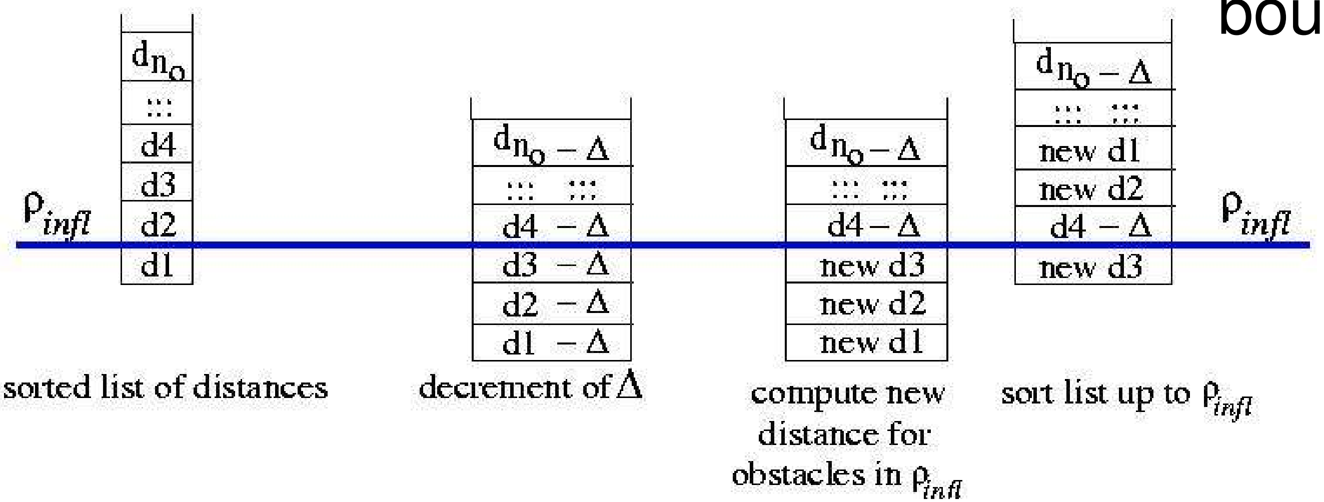
- between two configurations

  subtract Δ to all elements

  distance < ρ => recompute exact distance

  insert it in the list

- manage a sorted list of lower bounds



Experimental Results

- Implementation on several robots:
  - hilare2 (with trailer), cycab (car-like), dala (rover)
- Time of computation gain:
  - collision checking: divided by 10
  - gradient of potential: divided by 6

Conclusion

- Gradient of potential in configuration space:
  - no closed-form expression if the potential is a function of the distance to obstacles
- Filter useless robot-obstacle interactions pairs:
  - maximum influence distance of interactions
  - spatial coherence
- Future work: complexity of the filtering algorithm