Planning Strategies

RSS II Lecture 6
September 21, 2005

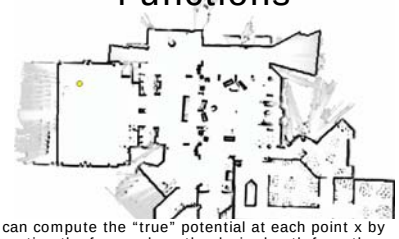
Today

- Strategies
  - What do we want from the planner?
- Tools
  - Robust motion planning
  - Exploration algorithms

Planner Capabilities

- Motion Planning
  - How do we get from the hangar to a brick and back?
- Robust Motion Planning
  - How do we avoid getting lost?
- Localization Recovery
  - We're lost. How do we recover?
- Exploration
  - Our map is incomplete/wrong. How do we get more data to build a better one?

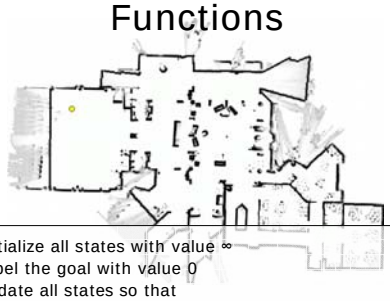
Numerical Potential Functions



- We can compute the "true" potential at each point x by integrating the forces along the desired path from the goal to x
- Let's use Dijkstra's algorithm, with the cost of an action given by

$$C(x) = F(x) = \nabla U_{att}(x) - \nabla U_{rep}(x)$$

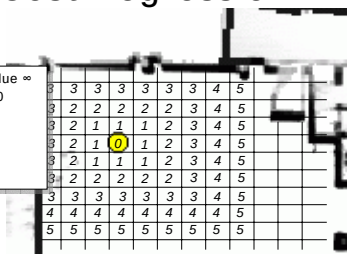
Numerical Potential Functions



1. Initialize all states with value ∞
2. Label the goal with value 0
3. Update all states so that f(x)=min(c(x,y)+f(y))
4. Repeat until all states labelled

Uniform Cost Regression

- Initialize all states with value ∞
- Label the goal with value 0
- Update all states so that f(x)=min(c(x,y)+f(y))
- Repeat

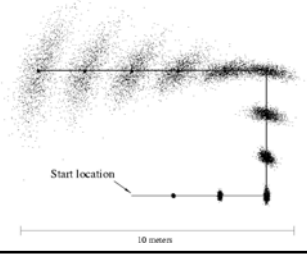


- After planning, for each state, just look at the neighbours and move to the cheapest one, i.e., just roll down hill

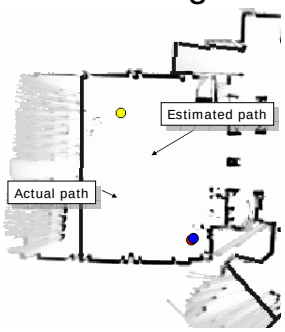
The Output Value Function

Motion is Uncertain

- Motion model is Gaussian about translation and rotation

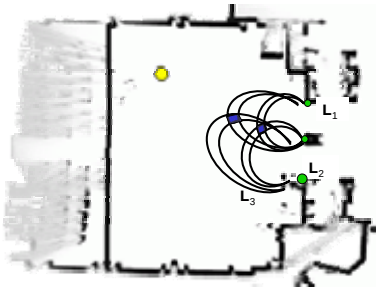


Brittle Navigation

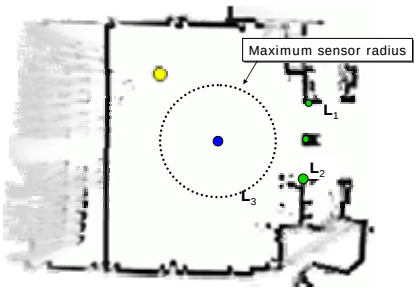


So what happened?

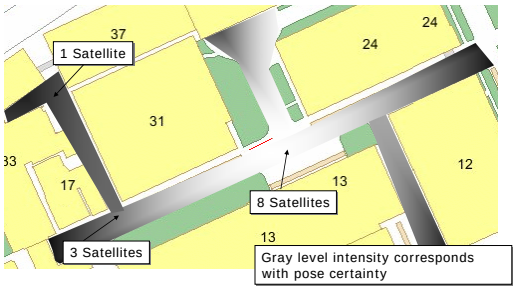
Sensors are Uncertain



Sensors can be blind



GPS and Uncertainty



Coastal Navigation

- Represent state using

Entropy: a probabilistic measure of uncertainty

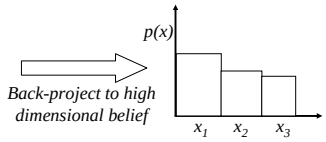
$$\tilde{x} = \left\langle \arg\max_s p(x), H(x) \right\rangle$$

1. Initialize all states with value -∞
2. Label the goal states with value f(y) = $E_y(r)$
3. Update all states so that f(x) = max(c(x,y) + f(y))
4. Repeat

Model Parameters

- Reward function

$R(\tilde{x})$

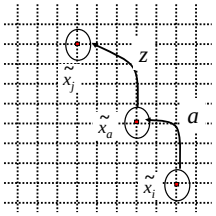


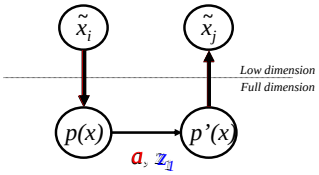
Compute expected reward from belief:

$$R(\tilde{b}) = E_b(R(s)) = \sum_s p(s)R(s)$$

Model Parameters

$f(\tilde{x}_i, a)$=?

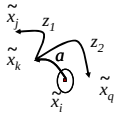




Deterministic and stochastic processes

Model Parameters

- Use forward model

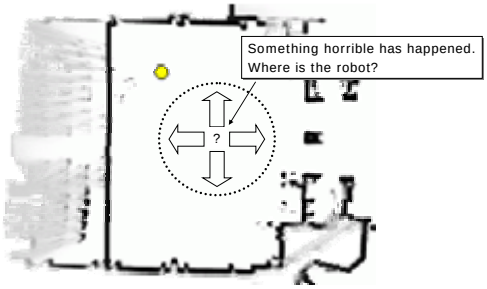


1. For each belief $\tilde{x}_i$ and action a
2. Generate full belief $p(x)$
3. For each observation z
   1. Compute $p(z|x, a)$
   2. Compute posterior $p'(x)$ and projection $\tilde{x}_j$
   3. Set $T(\tilde{x}_i, a, \tilde{x}_j)$ to $p(z|x_i, a)$

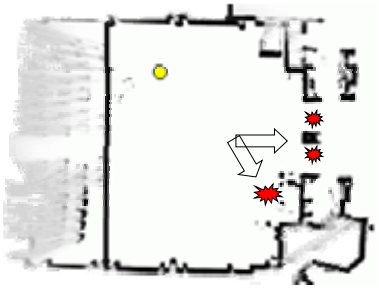
$$T(\tilde{x}_i, a, \tilde{x}_j) = \sum_{k=1}^{|Z_b|} \sum_{l=1}^{|X|} p(z_k \mid x_l) \sum_{m=1}^{|X|} p(x_l \mid x_m, a)\, p_j(x_m)$$

A Better Trajectory

Localization Recovery



Localization Recovery

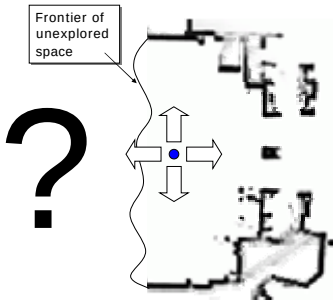


Localization Recovery

- If uncertainty is too large:
  - Hypothesize an action **a**
  - Change each possible position **p** to **p'** as if the robot were at **p** and took action **a**
  - Hypothesize what measurement you might get
  - Compute a new set of possible positions
  - Choose the action with the most certain posterior set of positions

Localization Recovery

Exploration



Exploration

- If pose uncertainty is small
  - Choose to visit nearest frontier
- Else
  - Use relocalization algorithm to reduce uncertainty

Exploration

- If there is a frontier that we
  - can visit,
  - but get no useful measurements in the unexplored space,
  - and still be able to return to the hangar
- Then
  - Visit that frontier
- Else
  - Use relocalization algorithm to first reduce uncertainty

Carmen Module APIs

- Who should decide what the planner should do?
  - Get blocks? Relocalize? Explore?
  - **Who should be in charge of the robot?**
- What knowledge should the module require?
  - Features? Distances? Landmarks? Directions? Maps?
  - Remaining power? GPS strength? Wireless network strength?
- What questions should module answer?
  - Can we get to a brick? How far? Will we get lost?
  - Can we find the hangar? With what confidence?
  - How many bricks can we collect?
  - Where are the unexplored regions?
- How should the planner integrate multiple sensor streams?
  - How should the planner handle vision data? GPS? Laser data? WiFi data?
- **Spiral development**
  - Put simple APIs in place, even if performance is stubbed

Conclusion

- Always use the full position estimate from the estimation algorithms
- Make decisions that respect current uncertainty
- Incorporate future uncertainty into plans