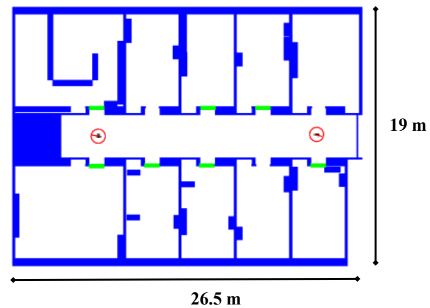
CSE-571
AI-based Mobile Robotics

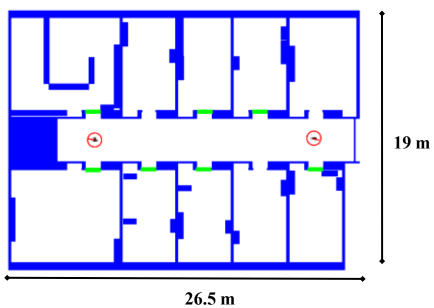
Active Sensing and Reinforcement Learning

Approximation of POMDPs: Active Localization

Localization so far: **passive** integration of sensor information



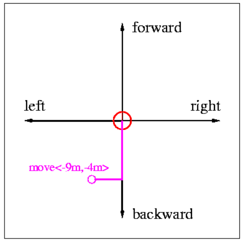
Active Localization: Idea

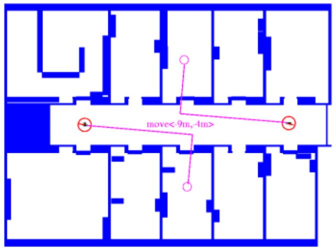


Efficient, autonomous localization by **active** disambiguation

Actions

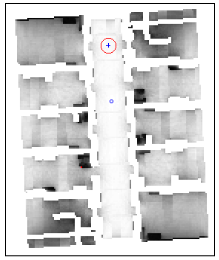
- Target point **relative** to robot
- Two-dimensional search space
- Choose action based on **utility** and **cost**





Utilities

- Given by change in uncertainty
- Uncertainty measured by entropy



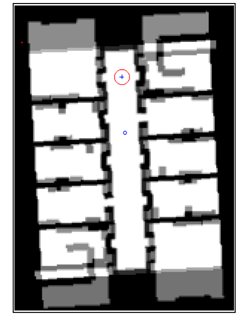
$$H(X) = -\sum_{x} Bel(x) \log Bel(x)$$

$$U(a) = H(X) - E_a[H(X)]$$
$$= H(X) + \sum_{z,a} p(z \mid x)\, Bel(x \mid a) \log \frac{p(z \mid x)\, Bel(x \mid a)}{p(z \mid a)}$$

Costs: Occupancy Probabilities

- Costs are based on occupancy probabilities

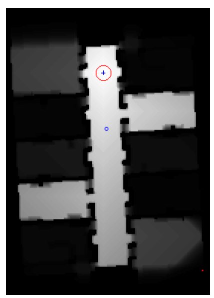
$$p_{occ}(a) = \sum_{x} Bel(x)\, p_{occ}(f_a(x))$$



Costs: Optimal Path

- Given by cost-optimal path to the target
- Cost-optimal path determined through value iteration

$$C(a) \leftarrow p_{occ}(a) + \min_{b} [C(b)]$$

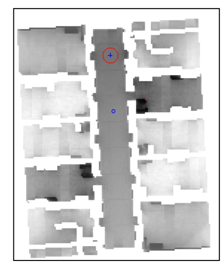


Action Selection

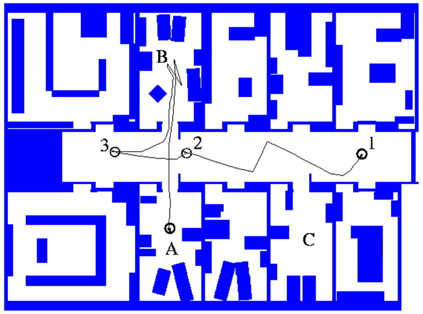
- Choose action based on expected utility and costs

$$a^* = \arg\max_{a} (U(a) - C(a))$$

- Execution:
  - cost-optimal path
  - reactive collision avoidance

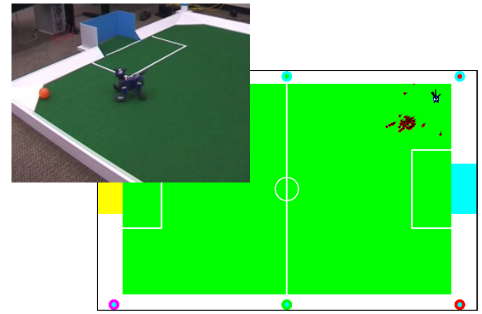


Experimental Results



- Random navigation failed in 9 out of 10 test runs
- Active localization succeeded in all 20 test runs

RL for Active Sensing



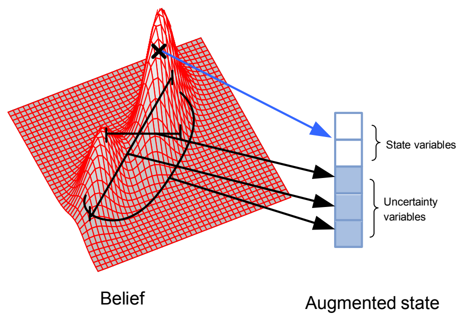
Active Sensing

- Sensors have limited coverage & range
- Question: Where to move / point sensors?
- Typical scenario: Uncertainty in only one type of state variable
  - Robot location [Fox et al., 98; Kroese & Bunschoten, 99; Roy & Thrun 99]
  - Object / target location(s) [Denzler & Brown, 02; Kreuchner et al., 04, Chung et al., 04]
- Predominant approach: Minimize expected uncertainty (entropy)

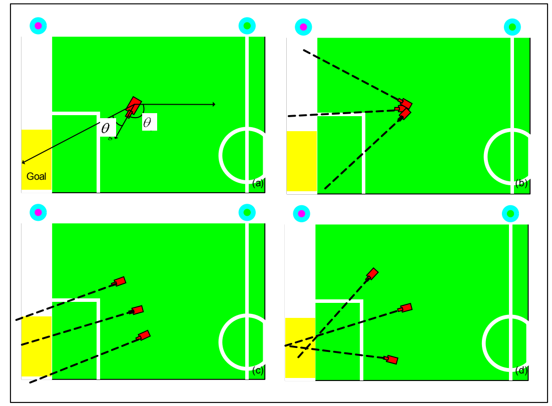
Active Sensing in Multi-State Domains

- Uncertainty in multiple, different state variables
  Robocup: robot & ball location, relative goal location, …
- Which uncertainties should be minimized?
- Importance of uncertainties changes over time.
  - Ball location has to be known very accurately before a kick.
  - Accuracy not important if ball is on other side of the field.
- Has to consider sequence of sensing actions!
- RoboCup: typically use hand-coded strategies.

Converting Beliefs to Augmented States



Projected Uncertainty (Goal Orientation)



Why Reinforcement Learning?

- No accurate model of the robot and the environment.
- Particularly difficult to assess how (projected) entropies evolve over time.
- Possible to simulate robot and noise in actions and observations.

Least-squares Policy Iteration

- Model-free approach
- Approximates Q-function by linear function of state features

$$Q^{\pi}(s,a) \approx \hat{Q}^{\pi}(s,a;w) = \sum_{j=1}^{k} \phi_j(s,a)\, w_j$$

- No discretization needed
- No iterative procedure needed for policy evaluation
- Off-policy: can re-use samples

[Lagoudakis and Parr '01,'03]

Least-squares Policy Iteration

$\pi \leftarrow \quad_0$

- Repeat

  $\pi \leftarrow$

  - Estimate Q-function from samples S

    $w^{\pi} \leftarrow \text{STD}\ Q(S, \gamma, \pi,$

    $\hat{Q}^{\pi}(s, a; w) \leftarrow \sum_{j=1}^{k} \phi_j(s, a)\, w_j$

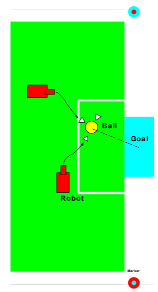
  - Update policy

    $\pi\ (s) = \text{arg}\max_{a \in} \hat{Q}^{\pi}(s, a, w)$

- Until ($\pi \approx \pi$)

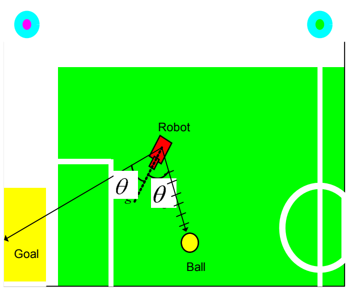
Application: Active Sensing for Goal Scoring

- Task: AIBO trying to score goals
- Sensing actions: looking at ball, or the goals, or the markers
- Fixed motion control policy: Uses most likely states to dock the robot to the ball, then kicks the ball into the goal.
- Find sensing strategy that "best" supports the given control policy.



Augmented State Space and Features

- State variables:
  - Distance to ball
  - Ball Orientation
- Uncertainty variables:
  - Ent. of ball location
  - Ent. of robot location
  - Ent. of goal orientation
- Features:

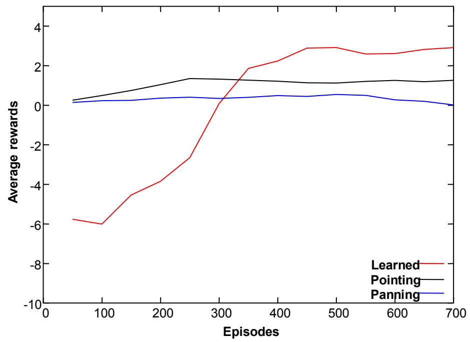


$$\phi(s, a, d_b) = \langle |\theta_b|, H_b, H_{\theta_g}, H_r, |\theta_a|, 1 \rangle$$

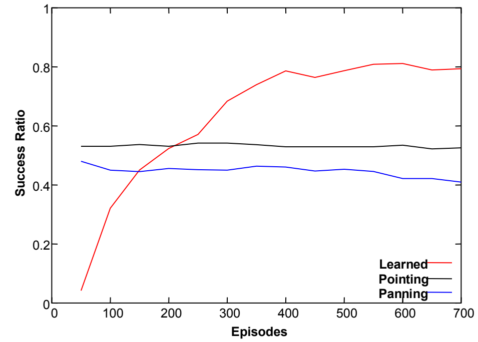
Experiments

- Strategy learned from simulation
- Episode ends when:
  - Scores (reward +5)
  - Misses (reward 1.5 – 0.1)
  - Loses track of the ball (reward -5)
  - Fails to dock / accidentally kicks the ball away (reward -5)
- Applied to real robot
- Compared with 2 hand-coded strategies
  - Panning: robot periodically scans
  - Pointing: robot periodically looks up at markers/goals

Rewards (simulation)



Success Ratio (simulation)



Learned Strategy

- Initially, robot learns to dock (only looks at ball)
- Then, robot learns to look at goal and markers

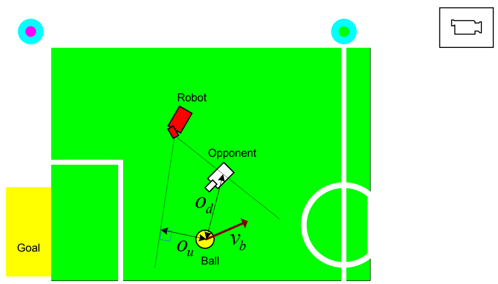
- Robot looks at ball when docking
- Briefly before docking, adjusts by looking at the goal
- Prefers looking at the goal instead of markers for location information

Results on Real Robots

- 45 episodes of goal kicking

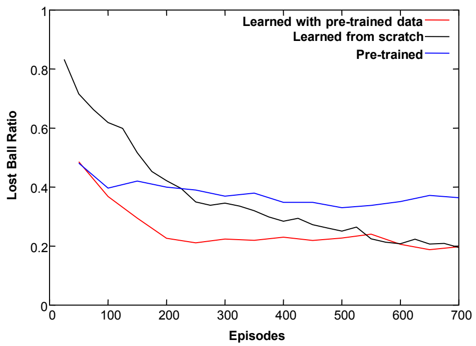
	Goals	Misses	Avg. Miss Distance	Kick Failures
Learned	31	10	6±0.3cm	4
Pointing	22	19	9±2.2cm	4
Panning	15	21	22±9.4cm	9

Adding Opponents



Additional features: ball velocity, knowledge about other robots

Learning With Opponents



- Robot learned to look at ball when opponent is close to it. Thereby avoids losing track of it.

Summary

- Learned effective sensing strategies that make good trade-offs between uncertainties
- Results on a real robot show improvements over carefully tuned, hand-coded strategies
- Augmented-MDP (with projections) good approximation for RL
- LSPI well suited for RL on augmented state spaces