Active perception using POMDPs

Matthijs Spaan

Institute for Systems and Robotics
Instituto Superior Técnico
Lisbon, Portugal

Reading group meeting, October 1, 2007
• Active perception: control your sensors.
• Context: URUS project.
  ► Network of cameras and robots.
  ► Outdoor, cameras won’t cover everything.
  ► Cameras detect events.
• Issues to tackle: modeling and solving.
Cooperative Perception Framework

• Static and mobile sensors.
• Using probabilistic feature models.
• Bayesian strategies to fuse uncertain information from spatially distributed sensors.
• Handling disagreement.
Active Perception

- Active perception will improve cooperative perception performance.
- Models interaction of active sensor with environment:
  - Point pan-and-tilt camera.
  - Choose more informative robot trajectory.
  - Ask robot to investigate area.
  - Execute expensive vision algorithm.
URUS scenarios

- **Navigation**
  - Adjust robot’s trajectory to improve perception.
  - Tradeoff information gain vs task execution.

- **Guiding**
  - Actively keep track of human.
  - Check whether human is still following.

- **Event detection**
  - Fixed camera detects possible event.
  - Ask robot to investigate.
Example scenario

- High level example scenario.
- Cameras detect fires and persons, noisy sensors.
- Robot moves to investigate.
- Fire occurs less, but are more valuable if detected correctly.
- Robot successfully reports events: trades off persons and fires.
• Partially Observable Markov Decision Processes (POMDPs).

• Models interaction with stochastic, partially observable environments.

• Probabilistic action and sensor models.

• Task: maximize long-term reward.
  ▶ Allows prioritization of objectives.
  ▶ Allows for actively reducing uncertainty.

• Solving is intractable: approximate methods.

• Decentralized POMDPs.
Beliefs

• Beliefs: probability distribution over states.
• Can be used to define quality of information.
• Common measure: entropy.
• POMDPs allow for trading off entropy minimization and task performance.
Solution methods

- Off-line vs. on-line (approximate) methods.
- Off-line methods such as “classic” POMDP solvers:
  - Rapid execution.
  - Off-line computation can be expensive.
- On-line methods, e.g., search the belief tree
  - Easily integrate belief-based rewards.
  - Require computation each decision moment.
- Are the POMDP models changing?
• Level of abstraction.
• Static or dynamic models.
• Solution method design.
• Communication issues.
• URUS integration.