On-line Coach and Markov Localization in the RoboCup Simulation League

João Pavão

Intelligent Systems Lab Workshop
Institute for Systems and Robotics

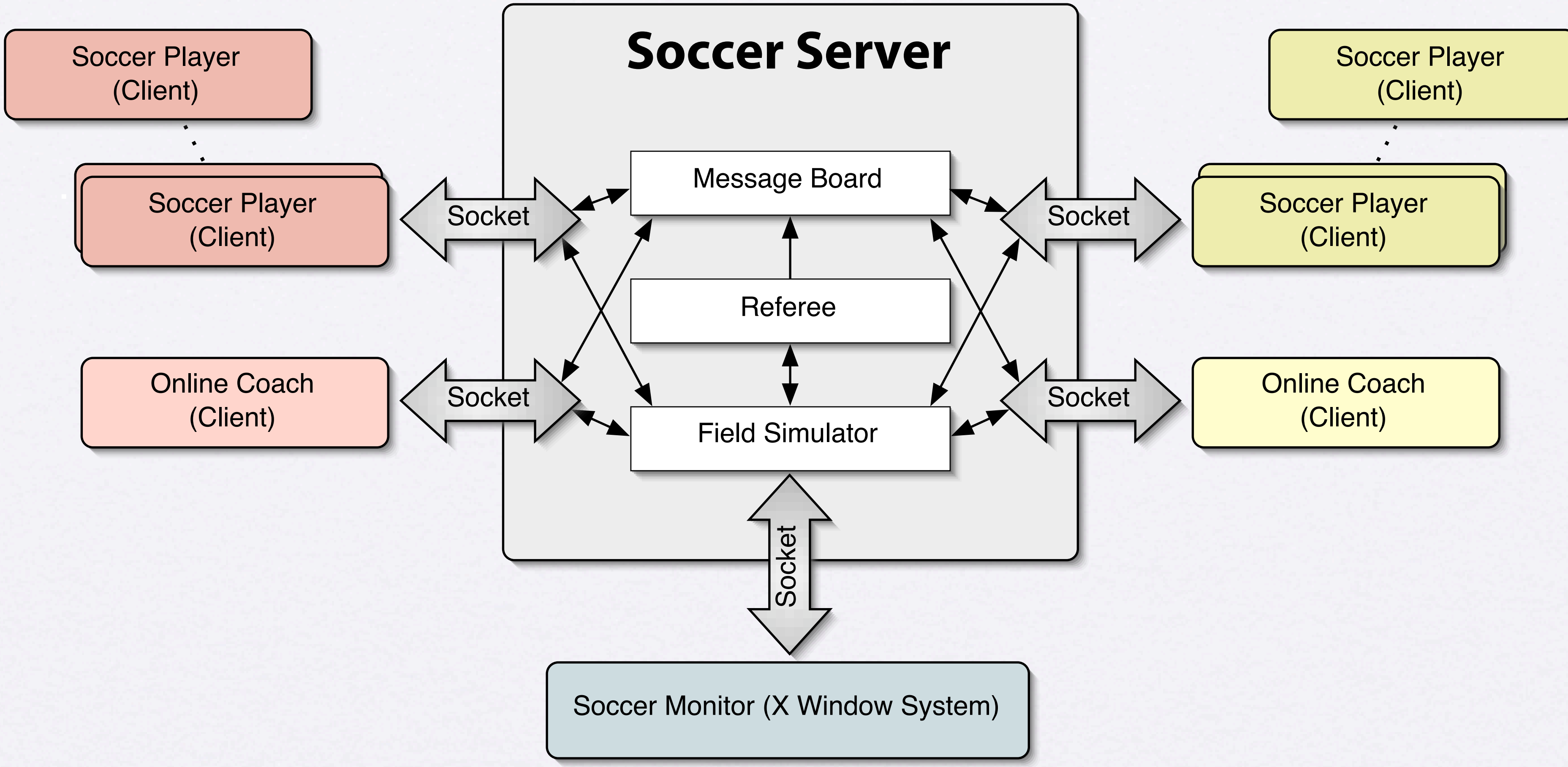
February 27, 2003

Presentation Outline

- On-line Coach:
  - The agent and its role in the Simulation System.
  - Agent capabilities.
  - CLang: the coach language.
  - Architecture overview.
  - Analysis techniques.
- Markov Localization

On-line Coach

Role in the Simulation System



The On-line Coach Agent

- No real-time demands
- Data delivered by *soccer server* is free from noise
- Limited communication
- Good tool for:
  - Opponent modeling
  - Game analysis
  - Delineating new strategies
- Has a competition of its own

Coach Agent Capabilities

- Limited audio communication:
  - Time interval between sent messages = 30 s
  - Delay after sent messages = 5 s
- Can make substitutions
- Receives perceptions: *look* and *hear*
- Sends actuation rules to its players

CLang: the Coach Language

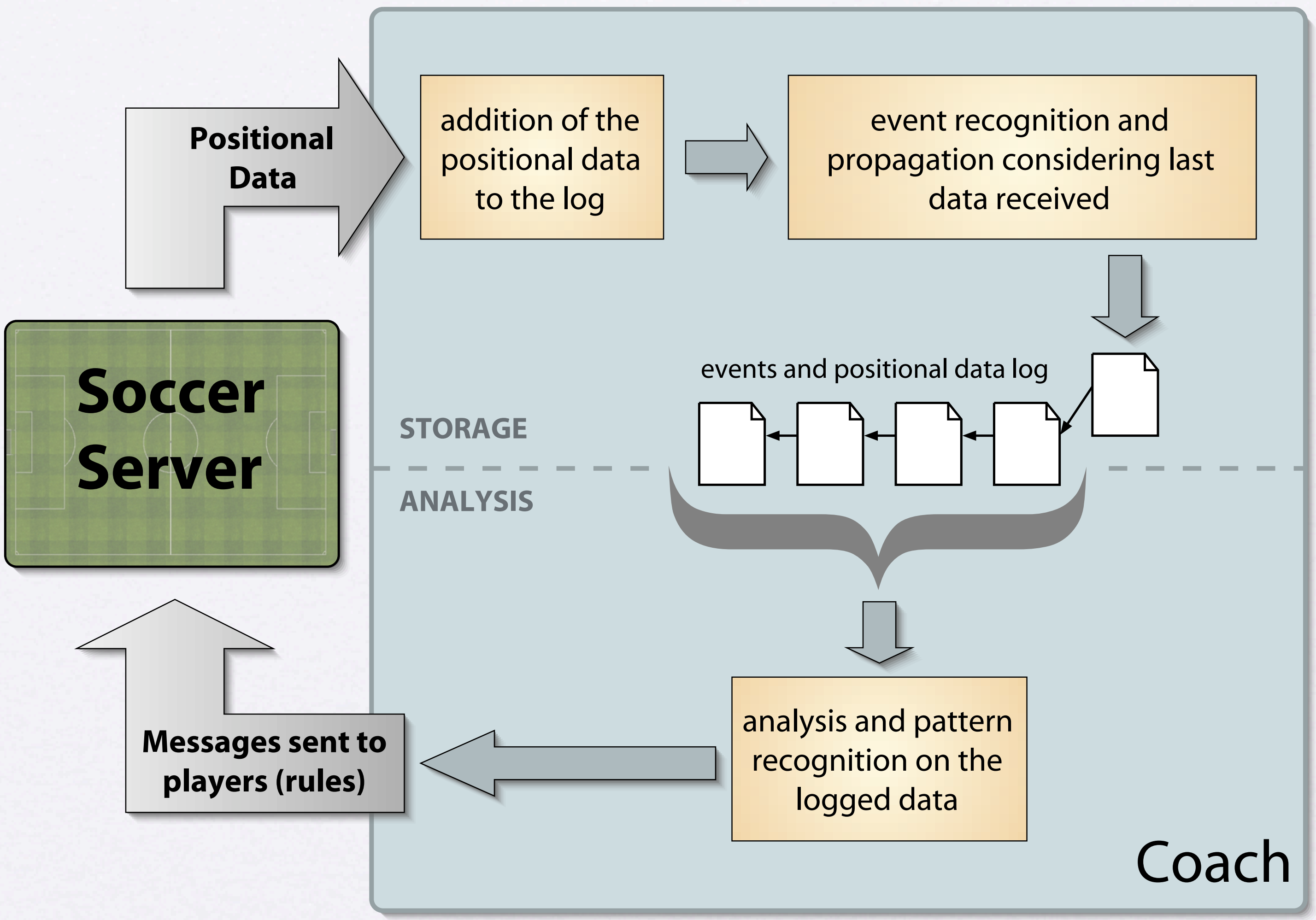
- Standard language defined by the RoboCup community

- Four different message types: *info*, *advice*, *define* and *meta*

- Messages are similar to *if-then* rules:

```
(info (6000 (playm ko_opp)
            (do opp {9} (bto {10}))))
```

Architecture Overview



Analysis Techniques

- Must produce useful results in a short time span.
- Must detect and react promptly to changes in tactics and formations of the opponent team.

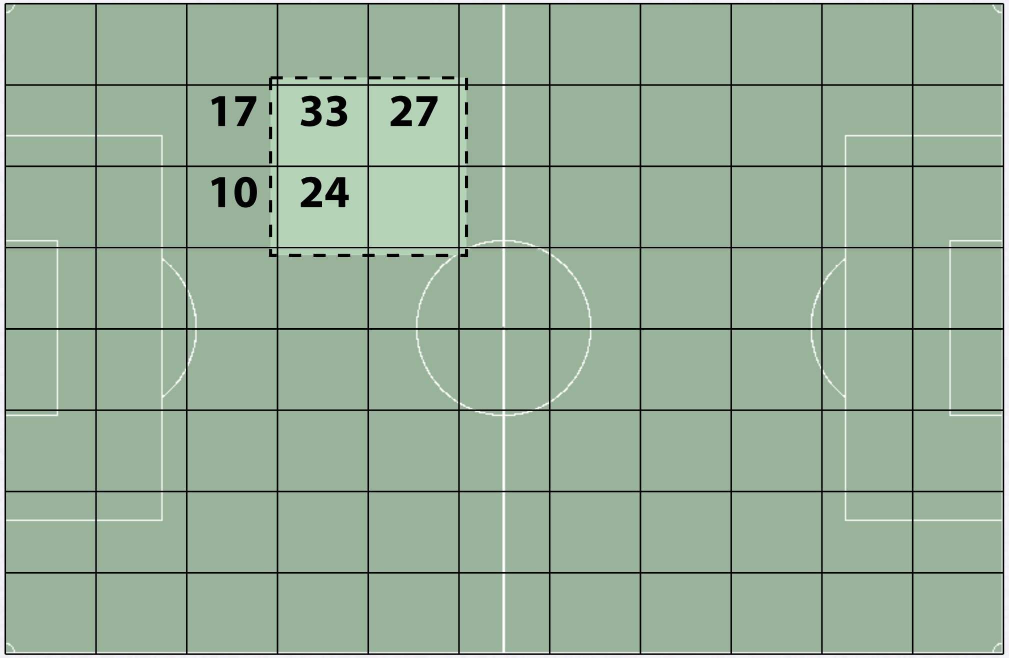
Analysis Techniques

1. Marking Opponents

- Classification of opponent players into *defender, midfielder* and *forward*.
- Differences in positions between and within groups of players.
- Pairing of our own defenders with forwards from the opponent team.

2. Adaptation to Positions of Opponents

- Tries to minimize distance to cover.
- Based on a grid for estimating opponent players probable location.



Soccer Playing Agents

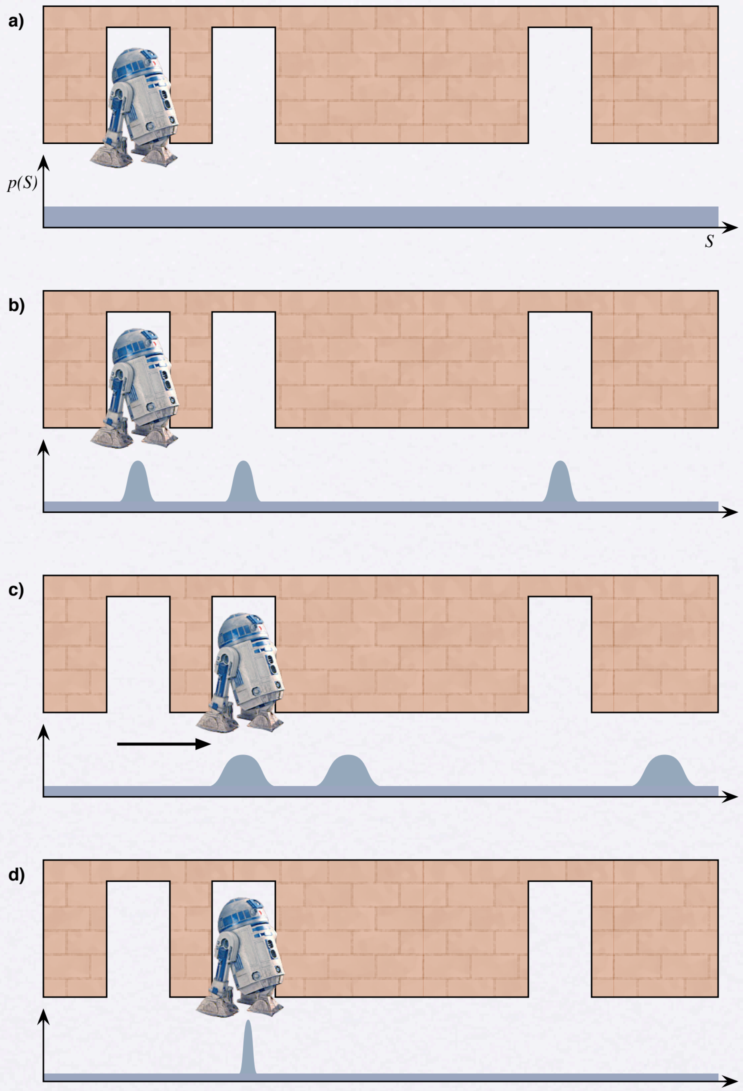
- Rule-based system with priorities
- The player has to *store* the rules sent by the Coach agent for later *retrieval*, *matching* and *selection*.

Markov Localization

Introduction

- Problem: given a model of the environment, estimate the location of the robot based on observations (odometric information and sensorial data).
- Global localization method – the robot is given a map of its environment (with landmarks) and has to estimate its position from scratch.
- Position probability grid approach

Classical Example



- Position probability density (belief) represented by multi-modal distributions.

$$Bel(L_t = l)$$

Algorithm

- General belief of being at location l at time t:

$$Bel(L_t = l) = P(L_t = l \mid d_{1,\ldots,t})$$

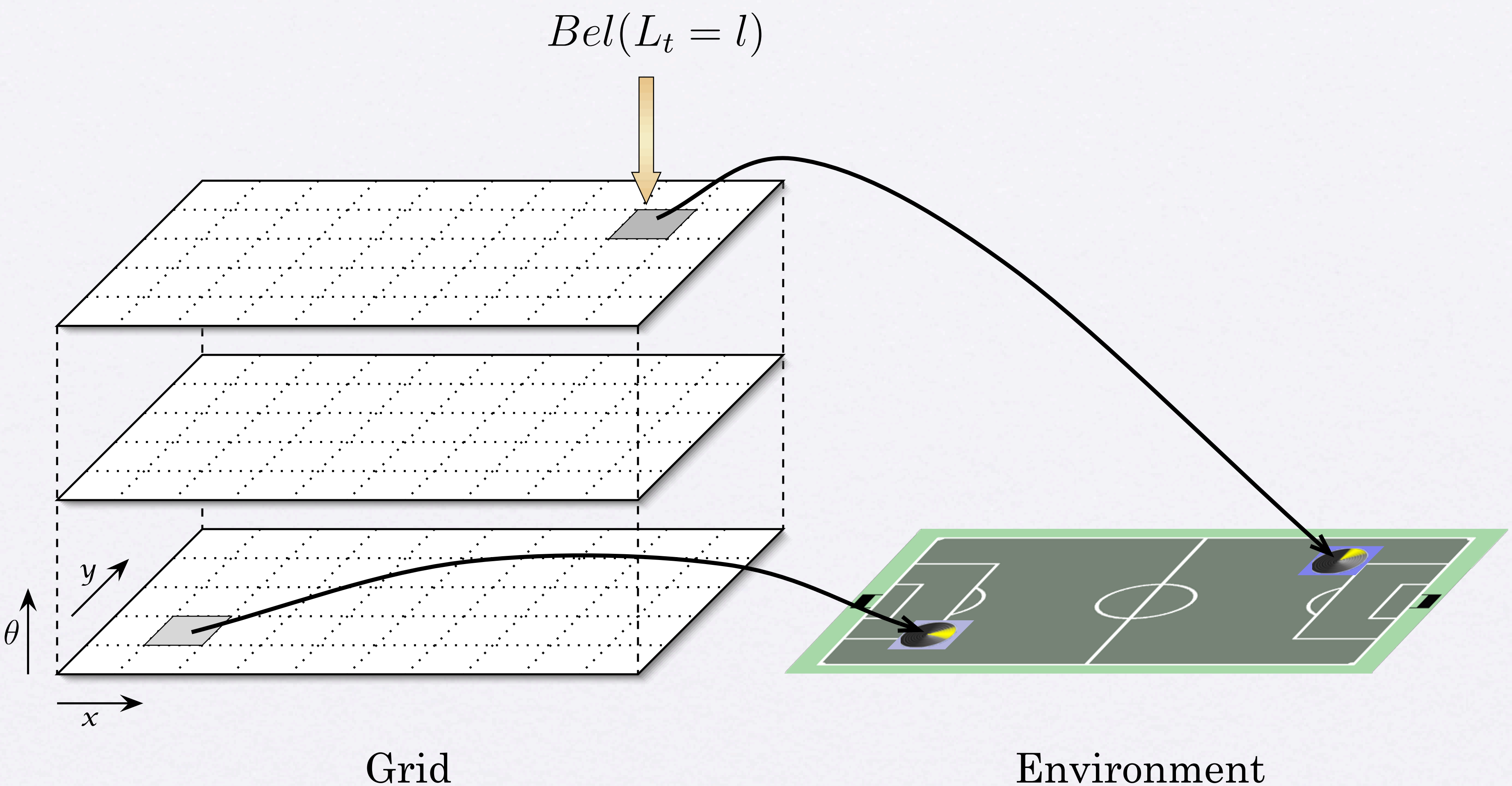
- Use recursive approximation instead, unfolded in two steps:
  - Prediction Phase (*motion model* → *odometry*):

$$P(L_t = l \mid L_{t-1} = l', a_{t-1}) = \frac{1}{\sigma(a_{t-1})\sqrt{2\pi}}\, e^{-\frac{(\mid l - l' \mid - a_{t-1})^2}{2\sigma^2(a_{t-1})}}$$

  - Update Phase (*sensor model* → *landmarks*):

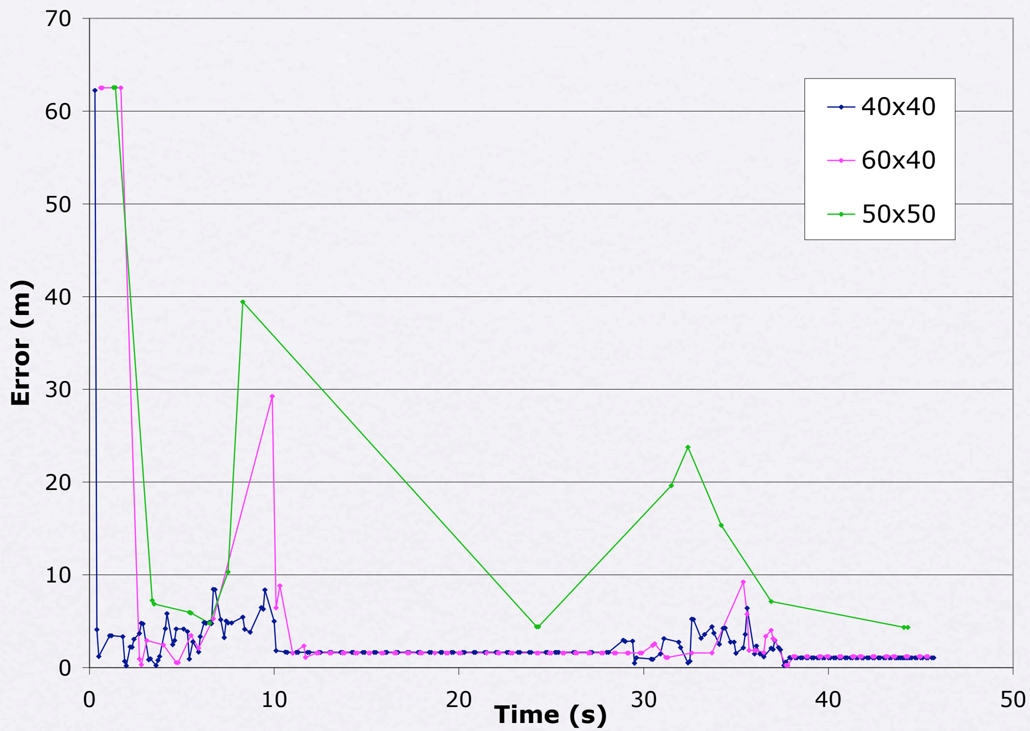
$$P(x) = \frac{1}{2\pi \mid \Sigma \mid^{1/2}}\, e^{-\frac{1}{2}\,(x-\mu)'\,\Sigma^{-1}\,(x-\mu)}$$

World Model

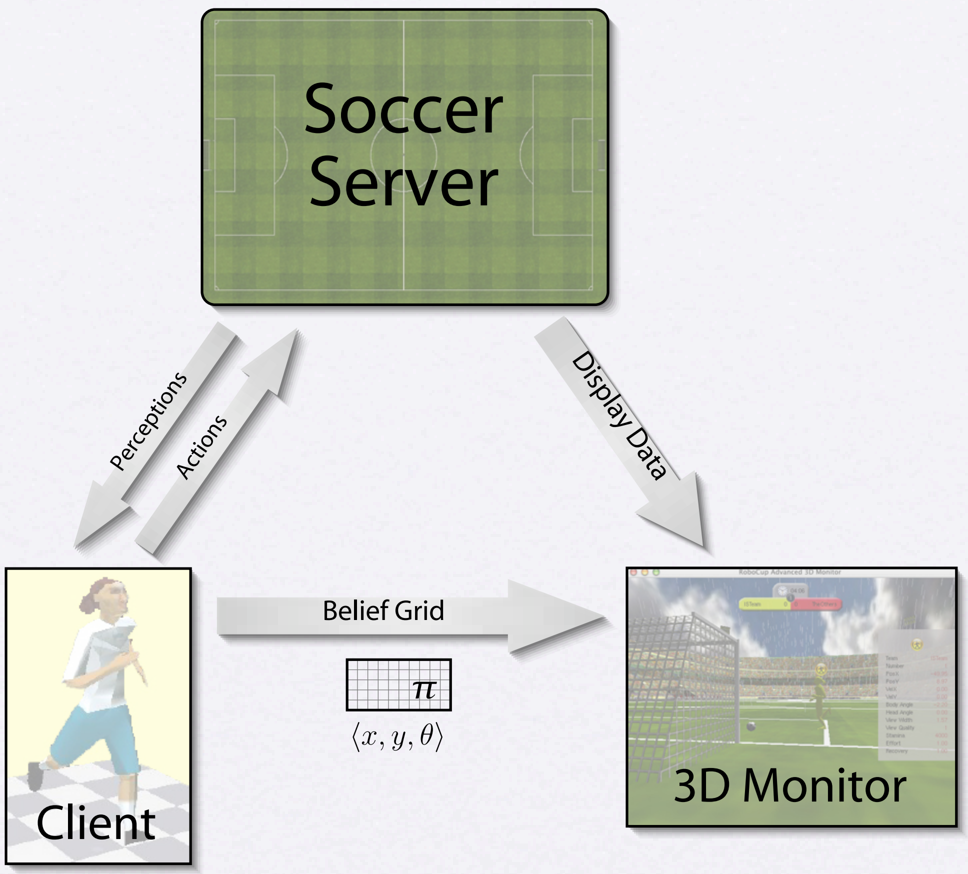


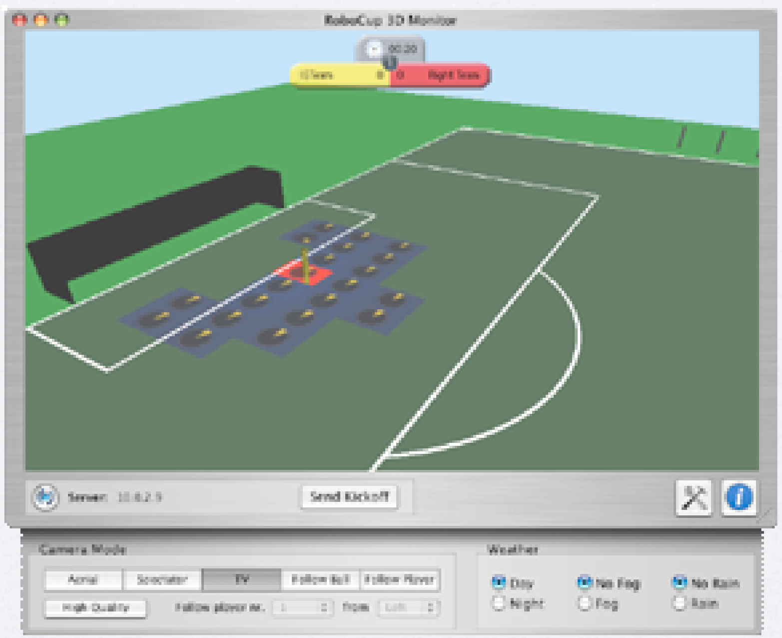
Experimental Results

Resolution	40 x 40	60x40	50x50
Cell Area (m)	2.63 x 1.70	1.75 x 1.70	2.10 x 1.36
Mean Pos. Error (m)	2.445	3.767	17.003



Integration with **RA3DM**





THE END