TeamChaos
Equipo Español de Fútbol Robótico

Humberto Martínez Barberá

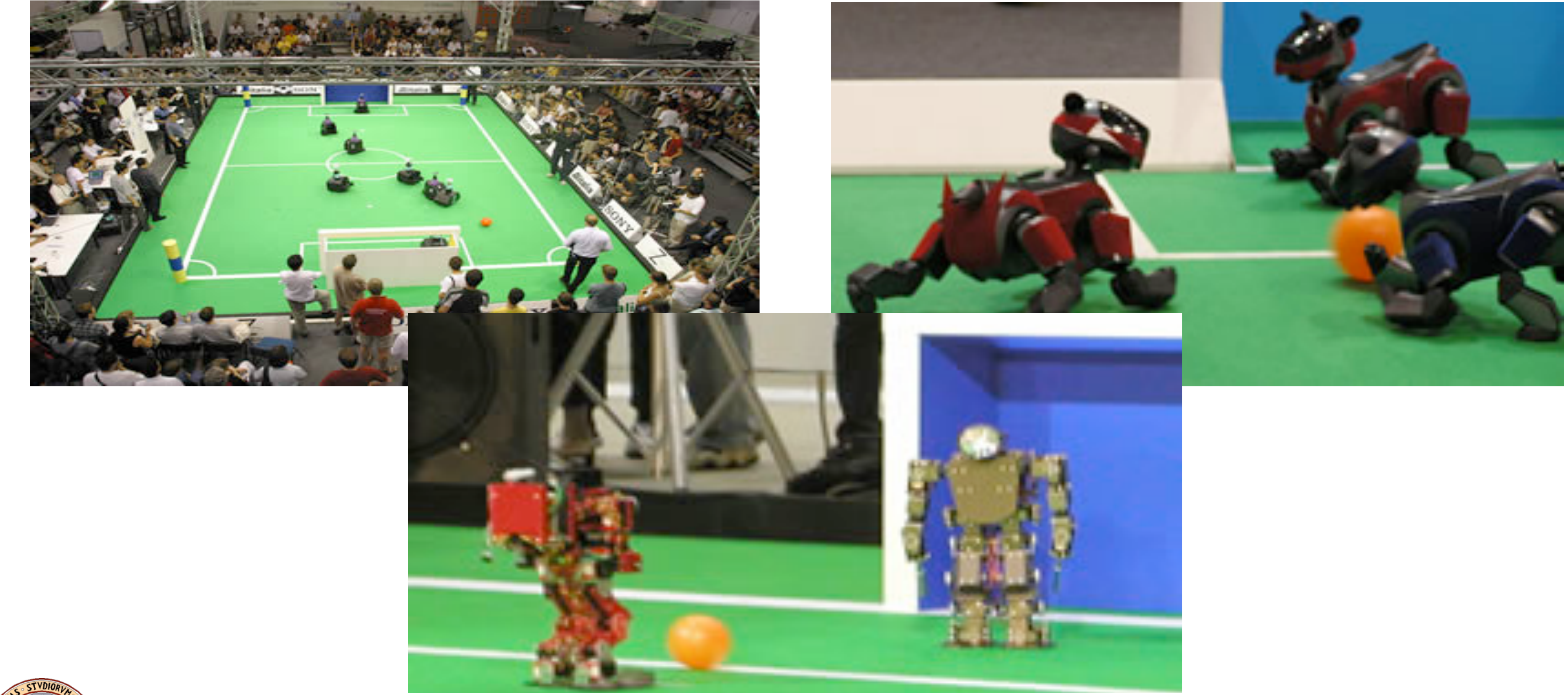
Dep. Ingeniería de la Información y las Comunicaciones

Universidad de Murcia

http://ants.inf.um.es/~humberto

The RoboCup

- **Goal**: By 2050, develop an autonomous humanoid team able to win the World Soccer Champion



The Four Legged League (I)

- The players: Sony AIBO robots
  - 3 DoF per leg
  - 3 DoF for the head
  - 1 CCD sensor
- The field
  - 2 coloured nets
  - 4/6 coloured landmarks
  - 4 dogs each team
- The team: Team Chaos
  - Örebro University (Sweden)
  - Lund University (Sweden)
  - Ronneby University (Sweden)
  - University of Murcia (Spain)

The Four Legged League (2)

- Problem: Complex nature of the environment
  - Some sources of complexity
  - Unreliable sensor system
  - High uncertainty and imprecision for localisation
  - Limited onboard resources
  - Complex locomotion system
  - Very dynamic environment
  - Hidden objects

TeamChaos Architecture (1)

- Approach: TeamChaos Architecture
  - Based on the ThinkingCap architecture

•Lower layer

•CMD: abstract layer to the sensori-motor functionalities

•Middle layer

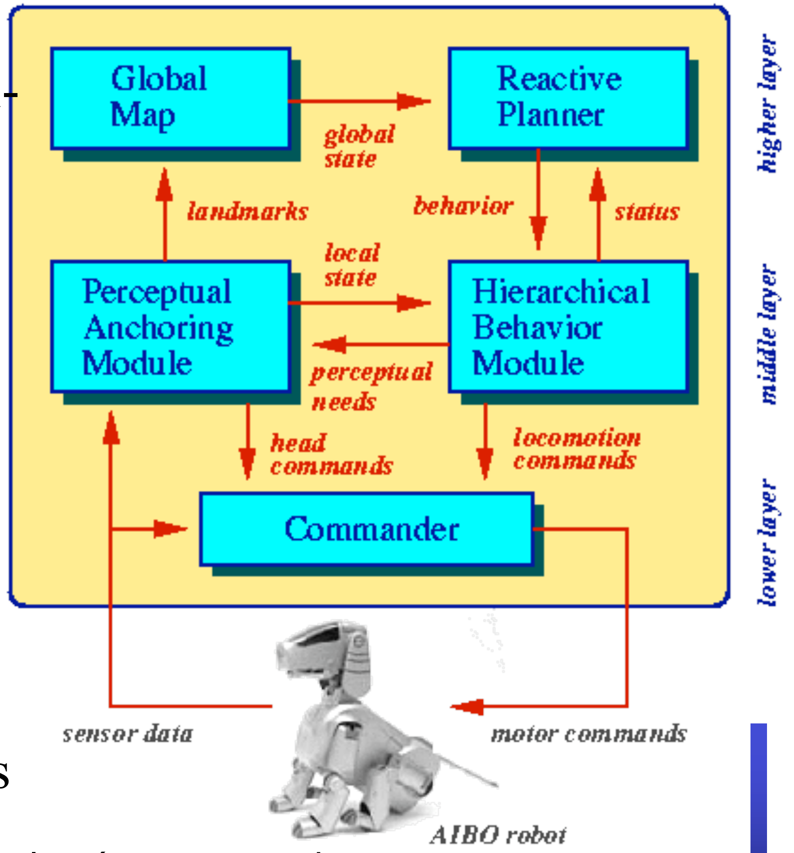
•PAM: consistent local representation around the robot

•HBM: set of robust tactical behaviours

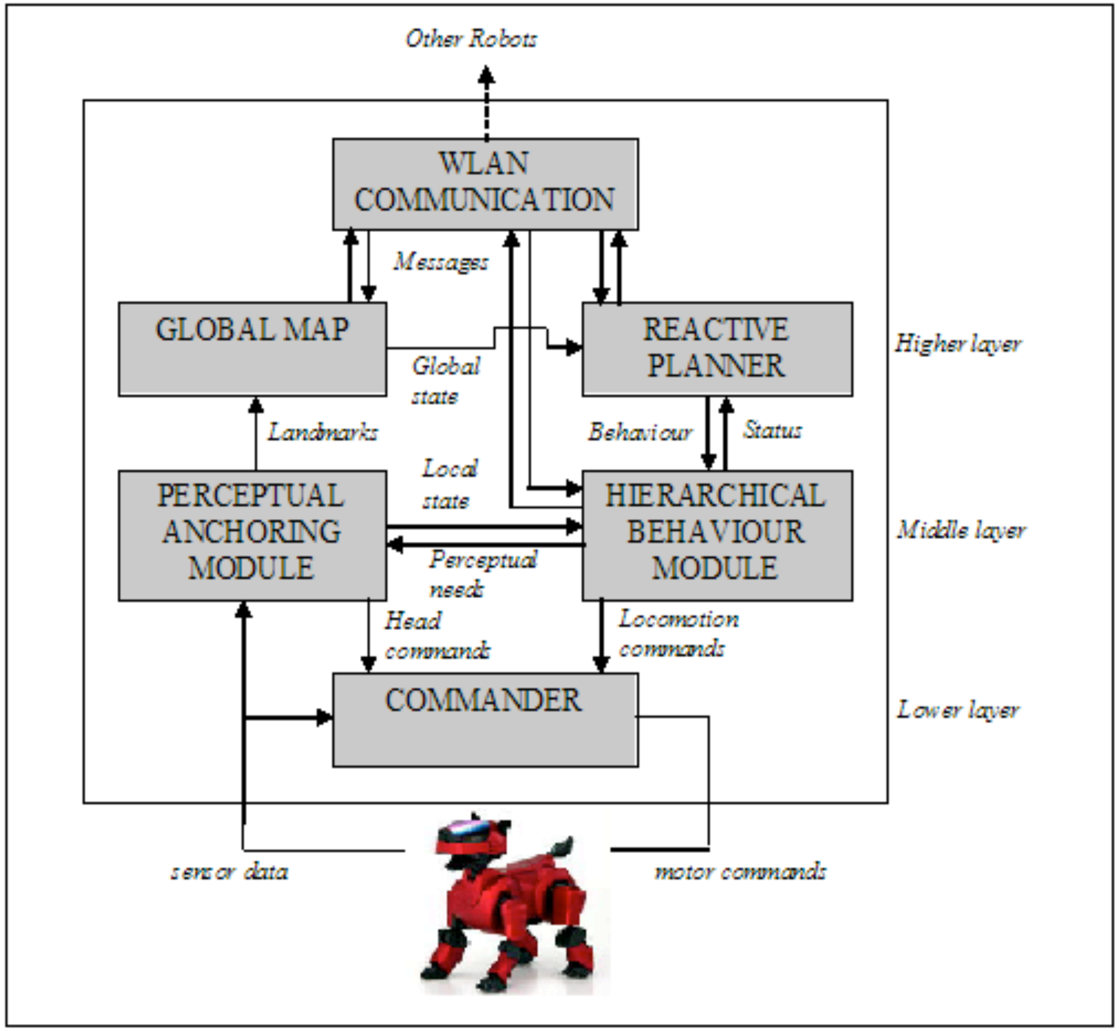
•High layer

•GM: global map of the field with ball information

•SAM: real-time strategic decisions

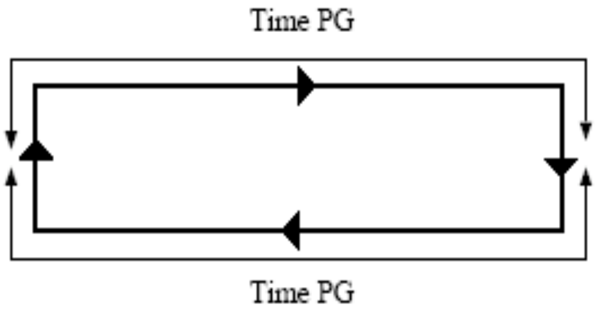


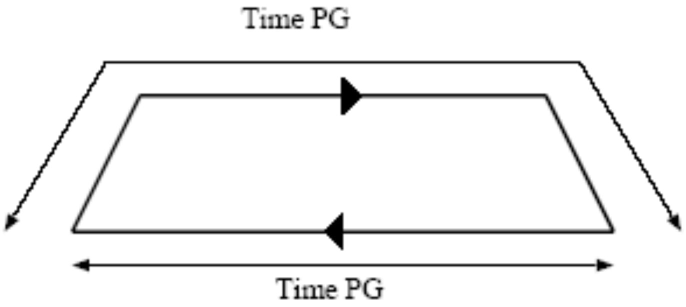
TeamChaos Architecture (2)

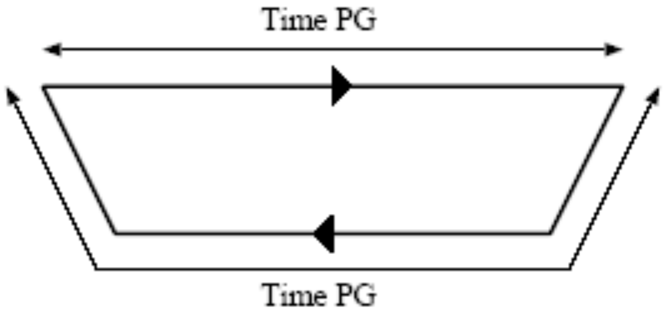


Commander

UNSW Walking Style (1)





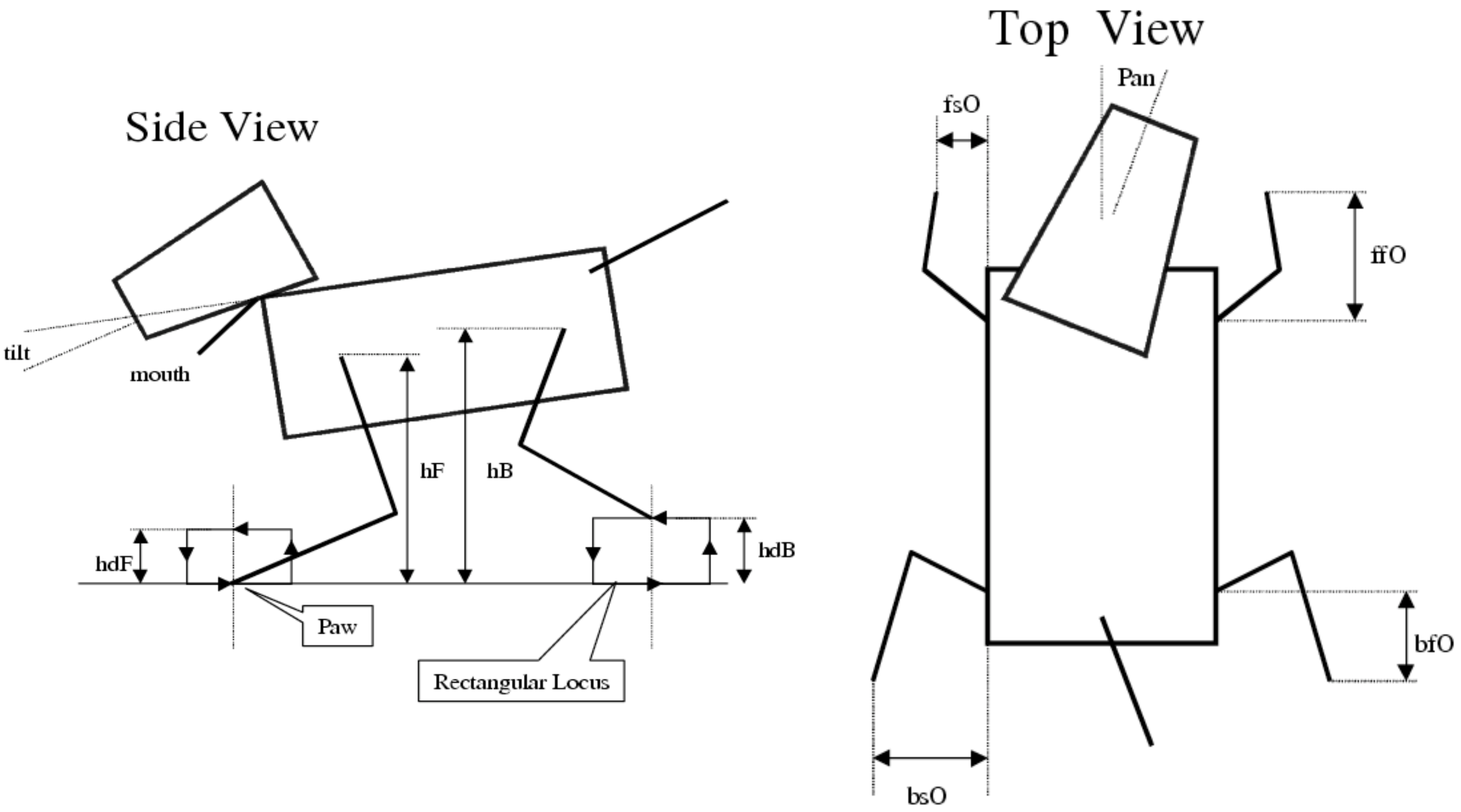


Current

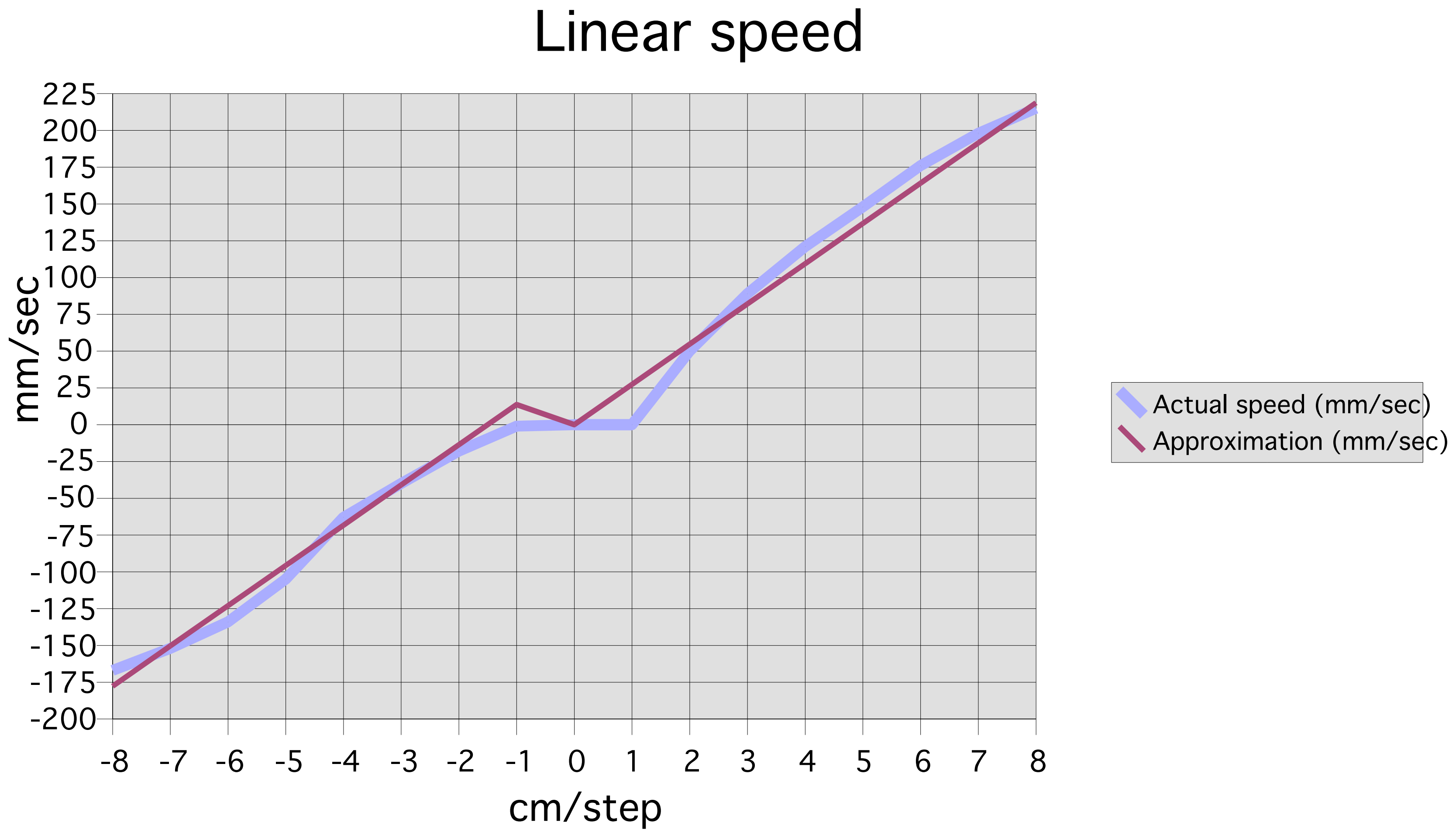
UNSW Walking Style (2)

- Advantages
  - Very low centre of gravity
    - The dogs practice a lot of body checking
  - Fast and stable gait
    - ERS-210: 300 cm/s (linear speed)
    - ERS-7: 400 cm/s (linear speed)
- Disadvantages
  - The dog presents a wider footprint
  - Lower pose makes difficult object viewing

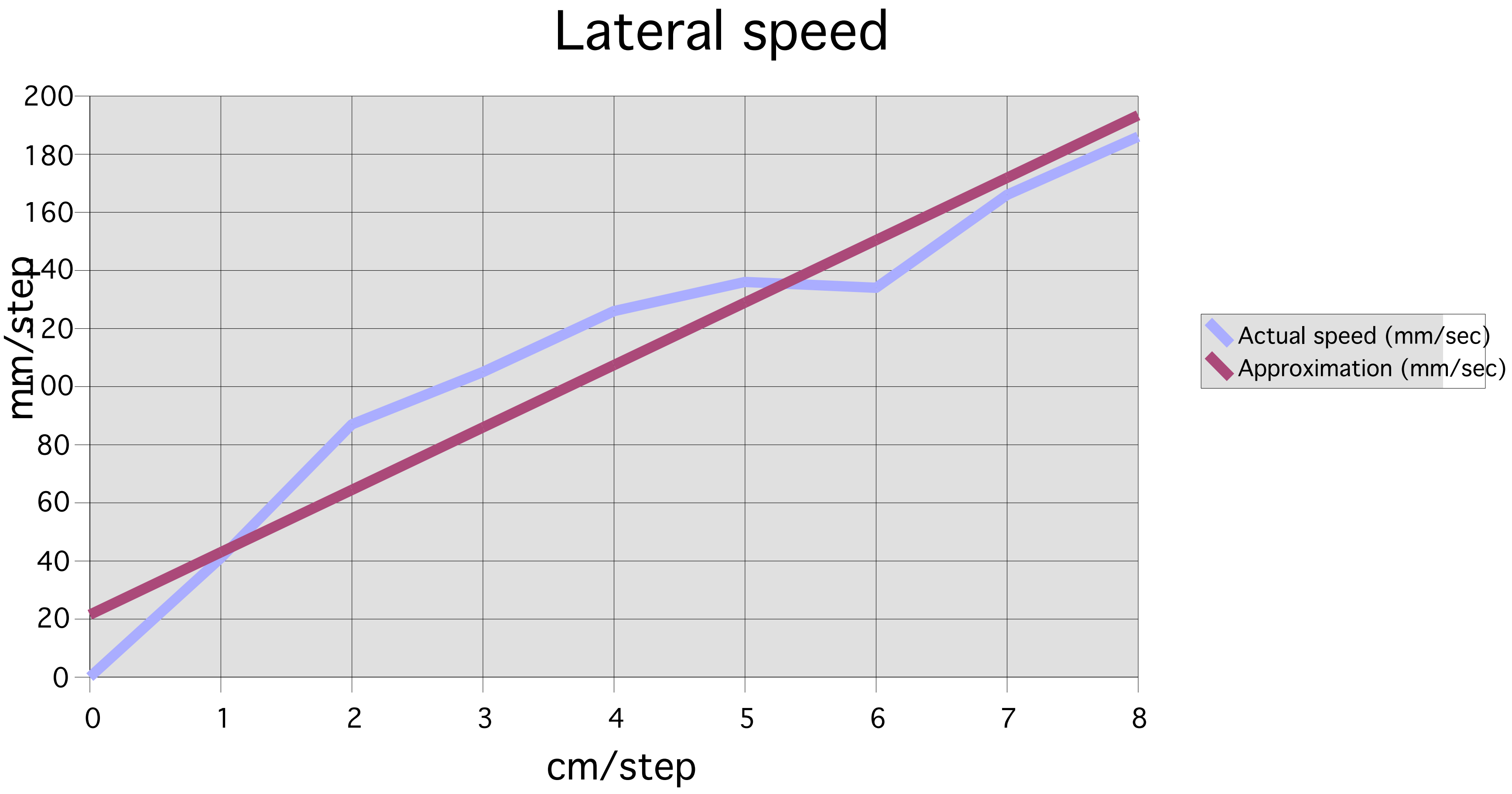
UNSW Model Parameters



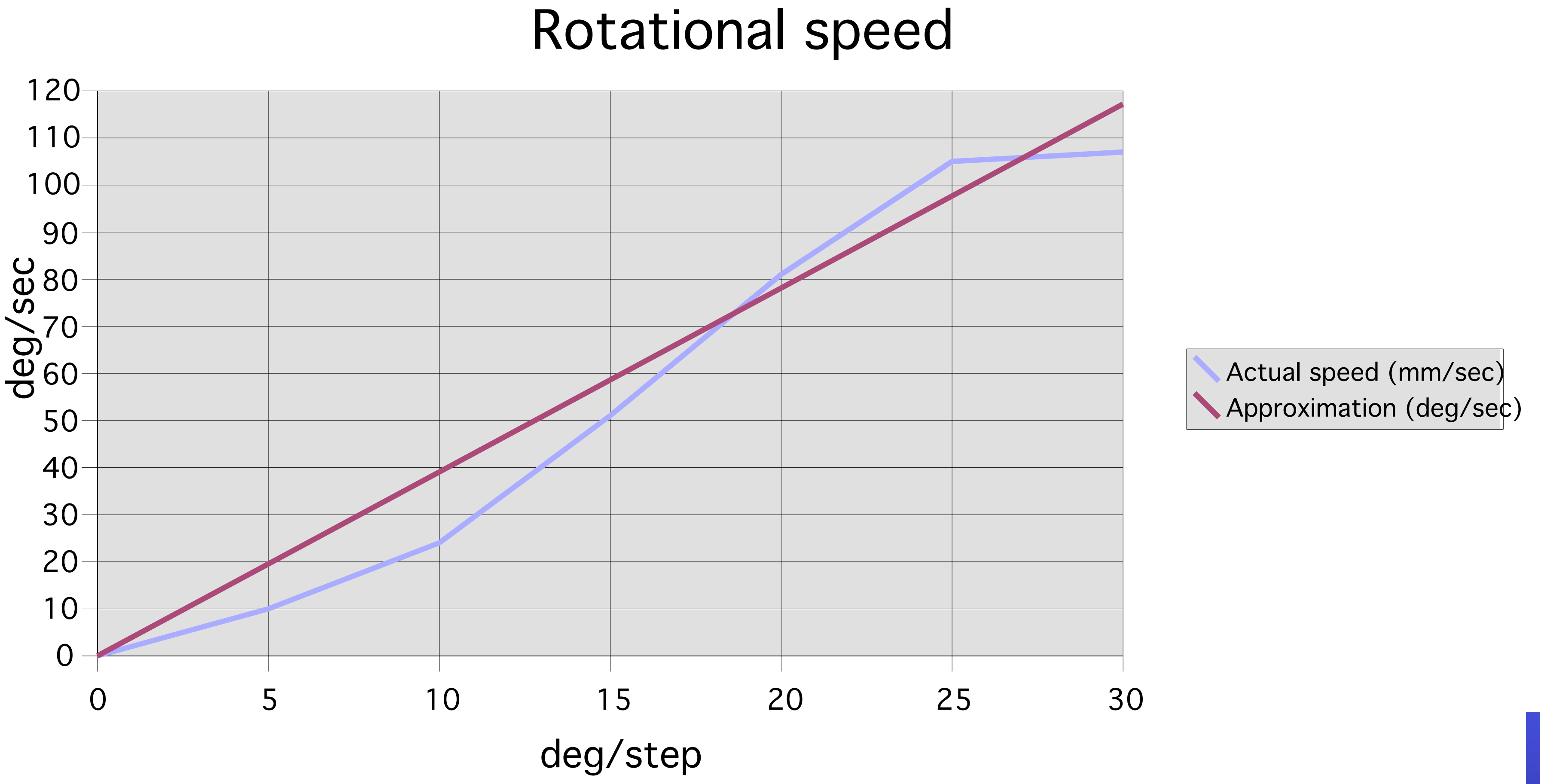
Odometry Calibration (1)



Odometry Calibration (2)



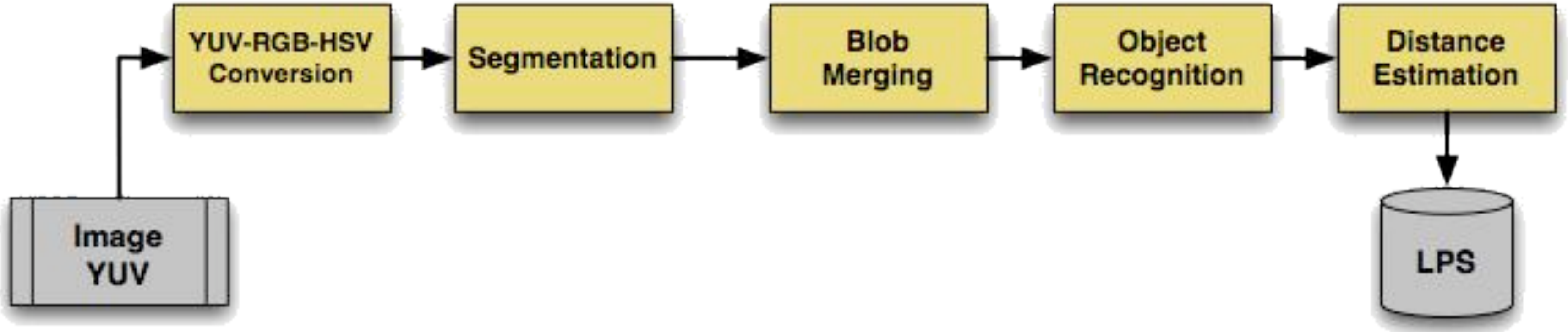
Odometry Calibration (3)



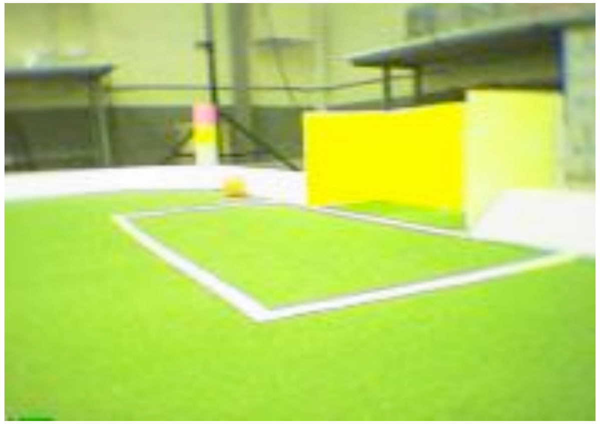
Perceptual & Anchoring Module

The Visual Pipeline

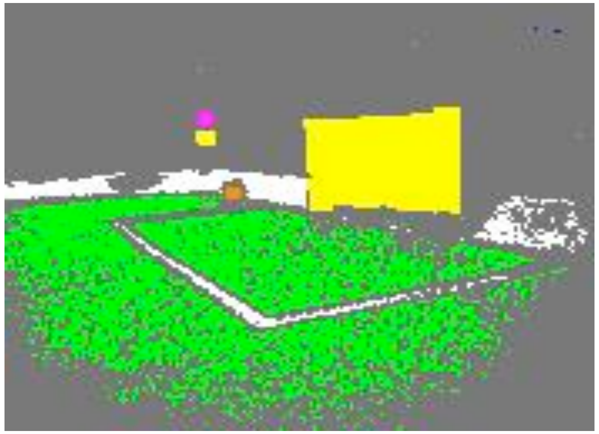
- Real-time color-based object detection (HSV)
  - Stable under different lighting conditions
  - Easy to calibrate
  - High resolution images
    - Does not use the robot CDT
  - Provides object coordinates in robot frame (ρ, θ)



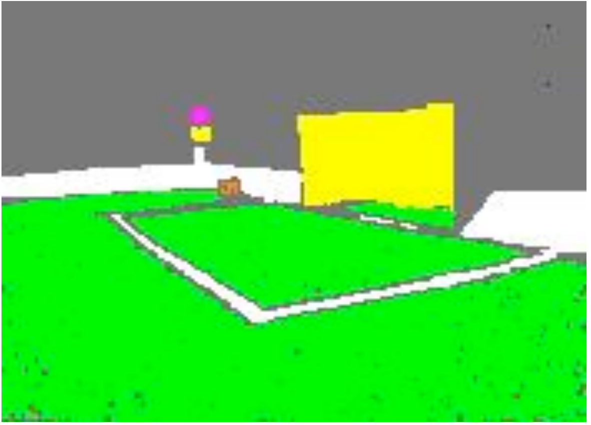
Examples



Raw Image



After Thresholding



Segmented Image

Approach for Feature Detection

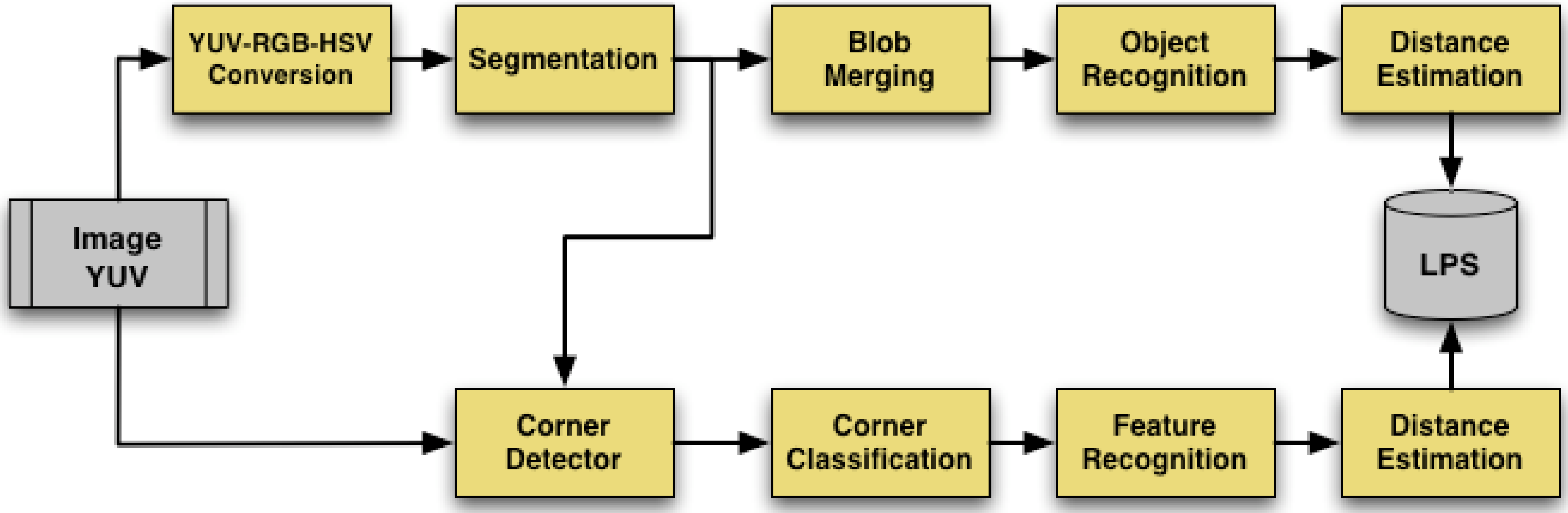
- The field presents some “natural” features
    - White field and zones lines.
- Why not lines?
    - Standard line recognition techniques are time-consuming
    - Lines all look the same
- We look for the intersections of lines, in a two-step process
    - Extract image corners produced by white lines
    - Group image corners into line intersection features
- Advantages of corner based features
    - They can be labelled (type T, type C, etc)
    - They can be fully tracked in a single image
        - Even with the small field of view of the AIBO camera
    - They can be computationally feasible and efficient

Corner Detection (1)

- We need to find corners in the images
  - Edge extraction and edge's corners detection
  - Gray level image processing
- We tried the second approach with two methods:
  - Based on brightness similarity (SUSAN)
    - [Smith and Brady, 1997]
  - Based on brightness gradient
    - [Sojka, 2002]
- Similar computational cost. The gradient-based one allows for more customization

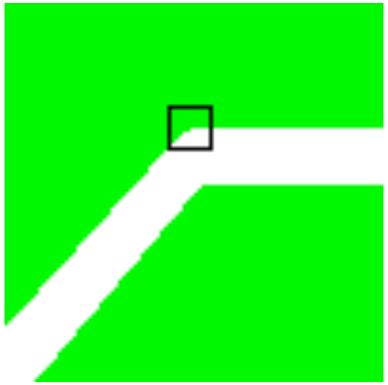
Corner Detection (2)

- The image corner detection is independent of the "standard" 4L-League objects after segmentation
  - Candidate corners are detected on the Y channel
  - Corners are filtered depending on HSV labels

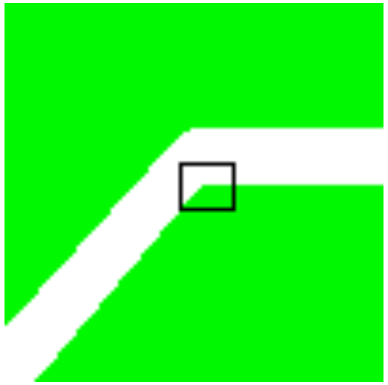


Corner Detection (3)

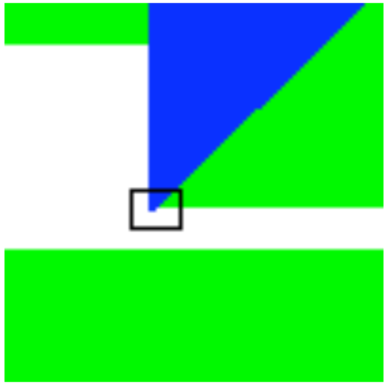
- The image corners are then classified into types
  - Open corner (convex): green and white
  - Closed corner (concave): green and white
  - Net corner: yellow/blue, green and white
- Other types of corners are rejected



Open



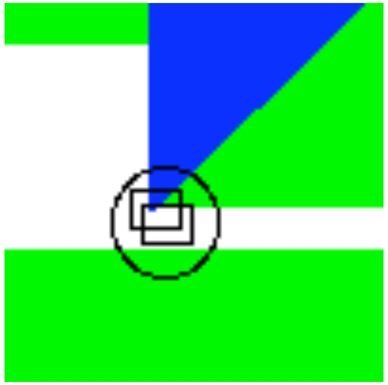
Closed



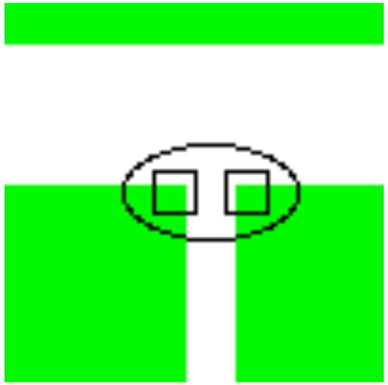
Net

Feature Recognition

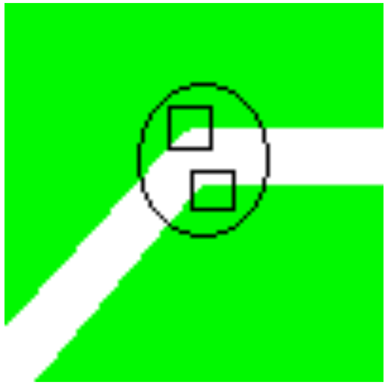
- Image corners are grouped into features
  - T net: one or more net corners (noisy)
  - T field: two closed corners
  - C field: an open and a closed corner
- There are different spatial constraints (distances)



T net

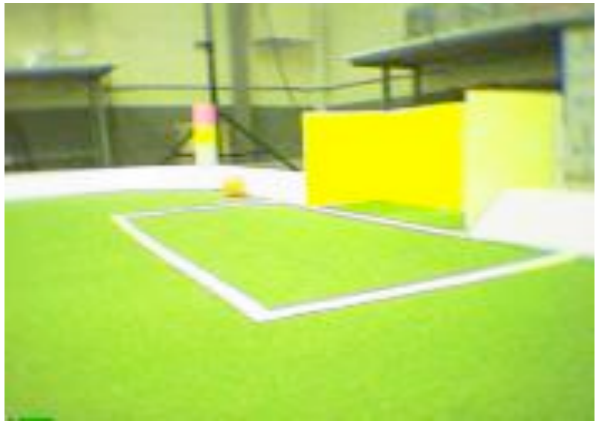


T field

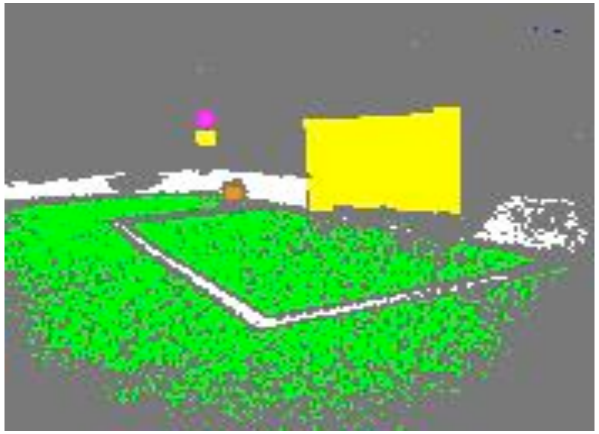


C field

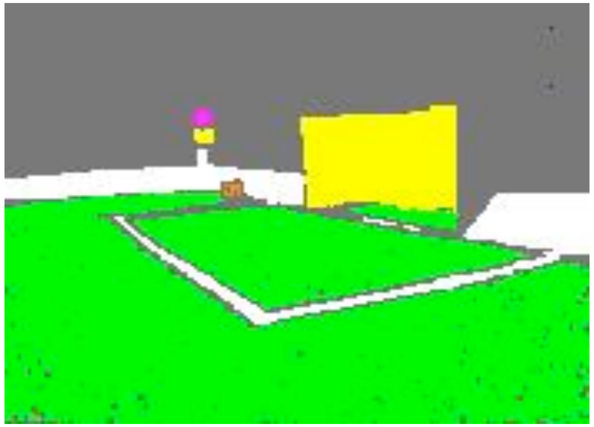
Examples (1)



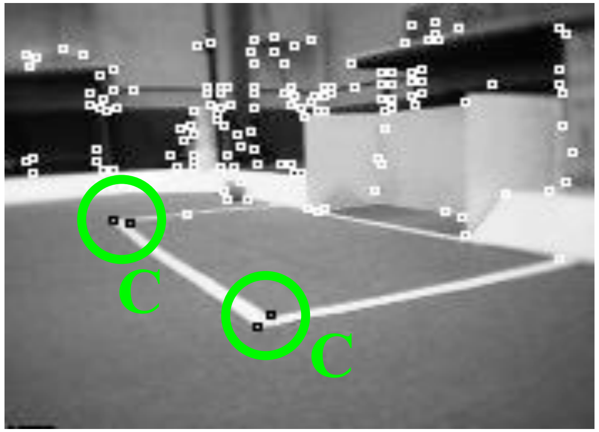
Raw Image



After Thresholding

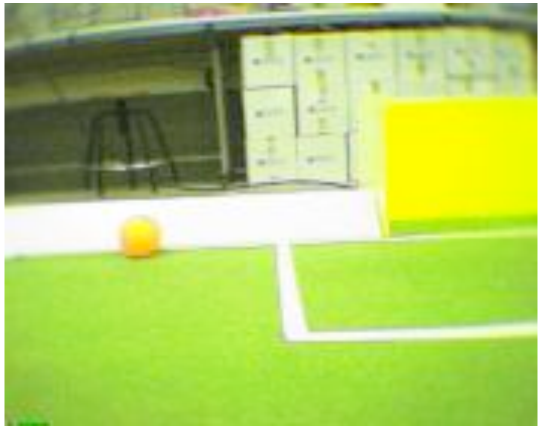


Segmented Image

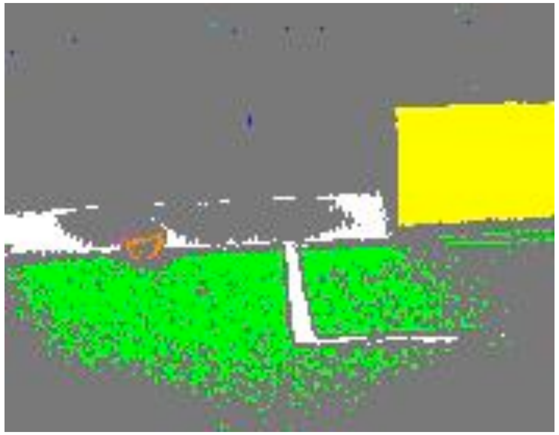


Detected Corners

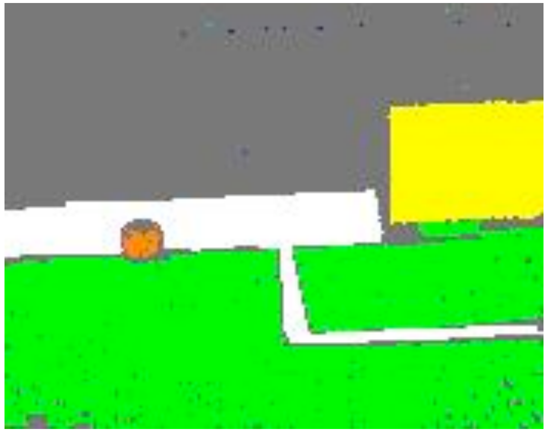
Examples (2)



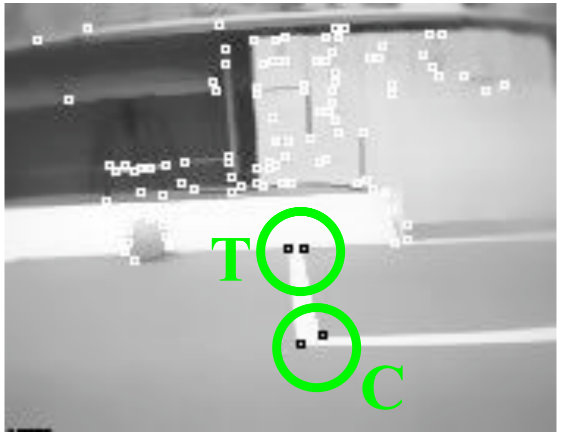
Raw Image



After Thresholding



Segmented Image



Detected Corners

Active Perception

- Automatic selection of the focus of attention
  - Based on the anchoring paradigm (PAM-HBM)
    - The controller sets which objects are needed
    - The perception sets which objects have been seen
  - Each object has an associated scan pattern
    - The ball is on the field (lower scans)
    - The landmarks are at 40 cm height (higher scans)

Hierarchical Behaviours Module

Behaviours Definition (1)

- HBM is implemented as a set of behaviours
  - Basic behaviours perform elementary types of actions
    - Many of them common to all players
    - Independent of the role of the player
  - Complex actions are obtained by hierarchical composition of behaviours
  - Behaviours are defined by way of fuzzy rules
    - Input space: estimates of objects from the PAM
    - Output space: velocity set-points <v_x, v_y, v_θ> and kick type <k> to the CMD

Behaviours Definition (2)

- Behaviours are composed using CDB (context-dependent blending)
  - Behaviours are coded with rules like:

    IF *predicate* THEN *action*

  - Behaviours are composed with rules like:

    IF *context* THEN USE *behaviour*

  - The fuzzy meta-rules have the advantages
    - Ability to express concurrent activation of behaviours
    - Smooth transition between behaviour

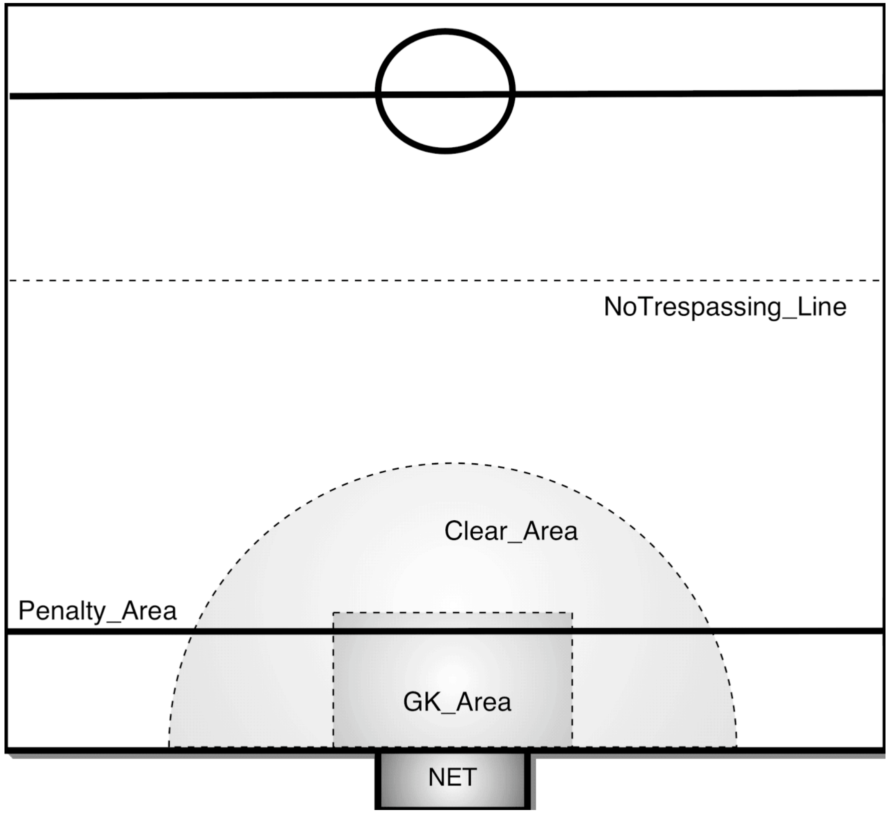
Goal Keeper (1)

- Task: to defend its net from possible scores
- We adopted a rather defensive strategy
  - Keeper stays close to its net
  - Keeper tries to be on the way between ball and net
  - Keeper only reacts toward the ball when there is a good opportunity to do so
- The robot operates in different modes
  - Localise: it tries to localise as best as possible, stays in the keeper area, and tries to face the ball
  - Defend: if the ball is in a risky position it tries to intercept it or kick it
  - Block: reflex which is activated when the ball is approaching the keeper

Goal Keeper (2)

- Strategy: the field is divided in different zones
  - Depending on the mode, the robot can reach different areas
  - Depending on the position of the ball, the robot decides different actions

- If the player is removed, because of a penalisation, it returns to the starting position avoiding other robots and the ball

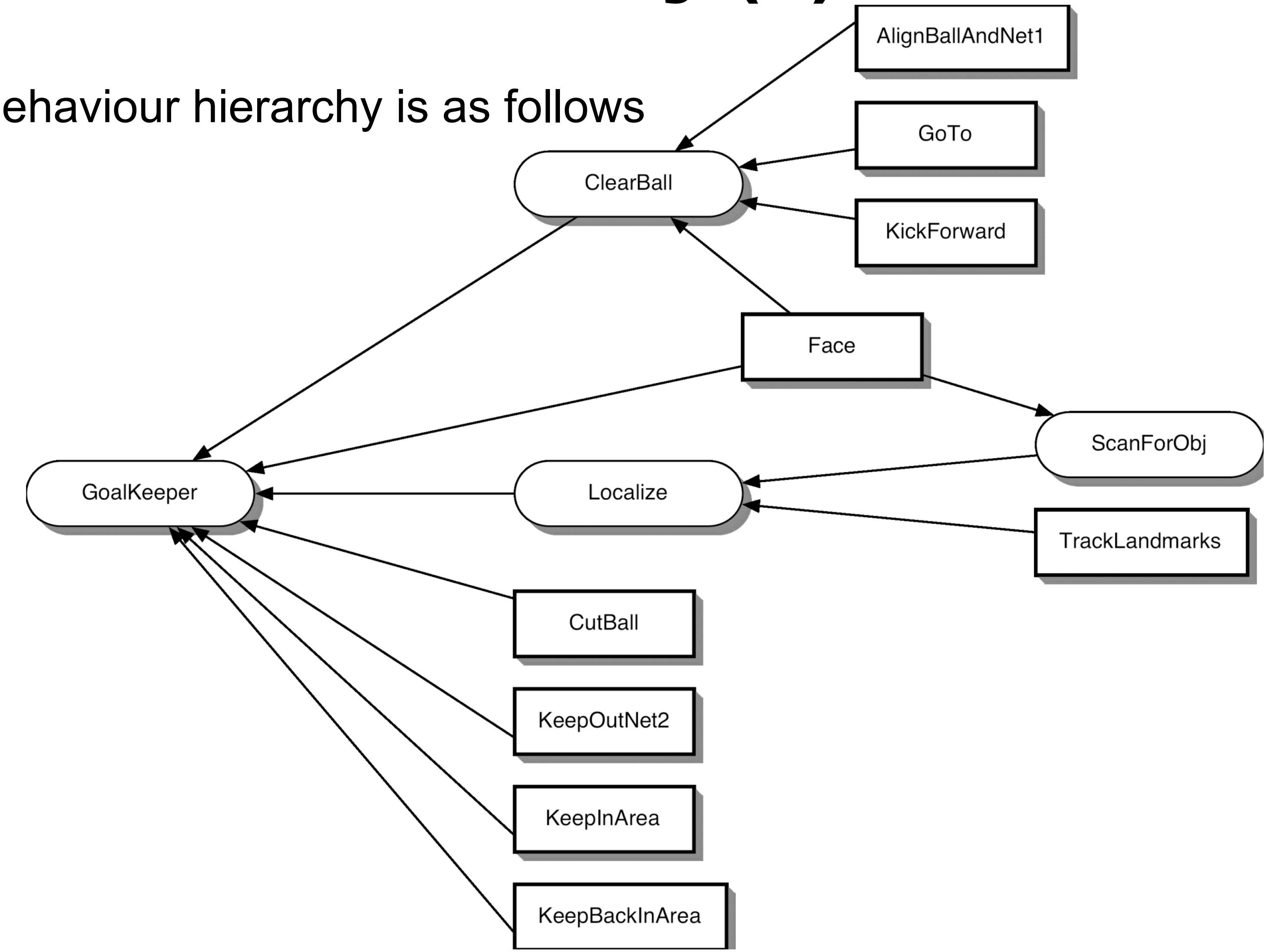


Behaviours Hierarchy (1)

- Three different types of behaviours
  - Simple common behaviours
    - Face: turn to directly face an object
    - GoTo: move to a given location
    - KickForward: move toward the ball and apply a kick
    - AlignBallAndNet1: move until it is aligned with the ball and the net
  - Simple specific goalkeeper behaviours
    - TrackLandmarks: select the least recently seen landmark as perceptual goal
    - KeepOutNet2: turn slowly until it is outside its own net
    - KeepInArea: turn to face forward and then move to its area
    - KeepBackInArea: turn to face backwards and then move to its area
    - CutBall: turn and move sideways to intercept the ball
  - Complex goalkeeper behaviours
    - GoalKeeper: top level goalkeeper behaviour
    - ScanForObj: turn on place until a given object is found
    - Localise: get a better position estimation by looking to the landmarks
    - ClearBall: turn and move forward to kick the ball

Behaviours Hierarchy (2)

- The behaviour hierarchy is as follows



Top Level Behaviour (1)

- Implemented as a Finite State Machine (FSM)
  - The action does not only depend on perception but also on the mode
  - Each mode is implemented as a set of fuzzy meta-rules

- Defend mode: reactively chooses a play strategy depending on the position of the ball
  - Predicate *ballClearable* evaluates if the ball is in Penalty_Area or Clear_Area
  - If the ball is not far the robot tries to reach it and then kick it
  - If the ball is far the robot tries to intercept the ball

state D

IF *ballClearable*	USE *ClearBall*
IF NOT *ballClearable* AND *inArea*	USE *CutBall*
IF NOT *ballClearable* AND NOT *inArea*	USE *KeepInArea*

Top Level Behaviour (2)

- Localise mode: tries to get a good estimate of the position to wait for the ball
  - There are two different sub-modes: when the position estimation is good (L_s) and when it is not (L_u)
  - Predicate *safeLocalised* evaluates the quality of the position estimation
  - If the estimation is good the robot tries to find the ball

state L_u

IF NOT *safeLocalised*	USE *Localise*
IF *safeLocalised* AND *inArea*	USE *Face* (Net1)
IF *safeLocalised* AND NOT *inArea*	USE *KeepInArea*

state L_s

IF NOT *safeLocalised*	USE *Localise*
IF *safeLocalised* AND *inArea*	USE *ScanForObj* (Ball)
IF *safeLocalised* AND NOT *inArea*	USE *KeepInArea*

Other Behaviours (1)

- The *CutBall* behaviour
  - Simple specialised behaviour
  - Only moves laterally and rotates. Does not move forward
  - If the ball is close and the robot is facing a safe area tries to kick

```
behaviour CutBall
IF posLeft                          SIDE (RIGHT)
IF posAhead                         SIDE (NONE)
IF posRight                         SIDE (LEFT)
IF headedLeft                       TURN (RIGHT)
IF headedAhead                      TURN (NONE)
IF headedRight                      TURN (LEFT)
IF freeToKick AND ballClose         KICK (FrontKick)
ALWAYS                              GO (STAY)
```

Other Behaviours (2)

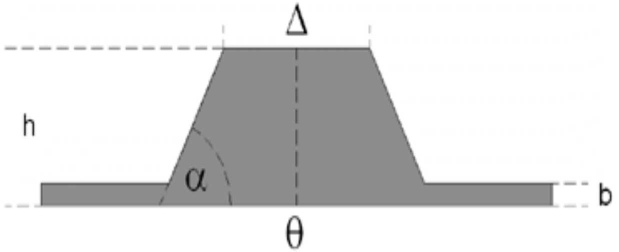
- The *ClearBall* behaviour
  - Hybrid behaviour
  - Turns until it faces the ball, moves the robot to approach the ball location, moves to align the ball and the opponent’s net
  - Depending on the orientation it uses different kicks

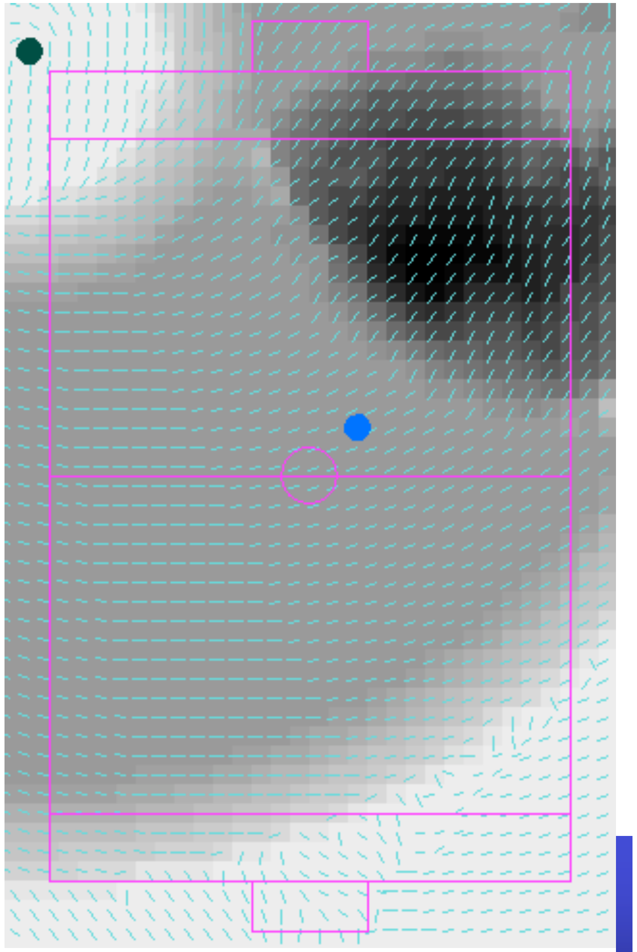
```
behaviour ClearBall
IF NOT ballTargeted                                   USE Face (Ball)
IF ballTargeted AND NOT ballClose                     USE GoTo (Ball)
IF ballTargeted AND ballClose AND freeToKick          USE KickForward
IF ballTargeted AND ballClose AND NOT freeToKick      USE AlignBallAndNet1
IF freeToKick AND freeLeftKick                        KICK (RightHeadKick)
IF freeToKick AND freeFrontKick                       KICK (FrontKick)
IF freeToKick AND freeRightKick                       KICK (LeftHeadKick)
```

Global Map

Fuzzy Position Grids

- Tessellation of space (3D). Cell $c_{ij} = (x, y, \theta)$
  - Possibility that the robot is at c_{ij}
  - In practice, $c_{ij} = (x, y) + \mu_\theta$
    - Where μ_θ = fuzzy set on θ

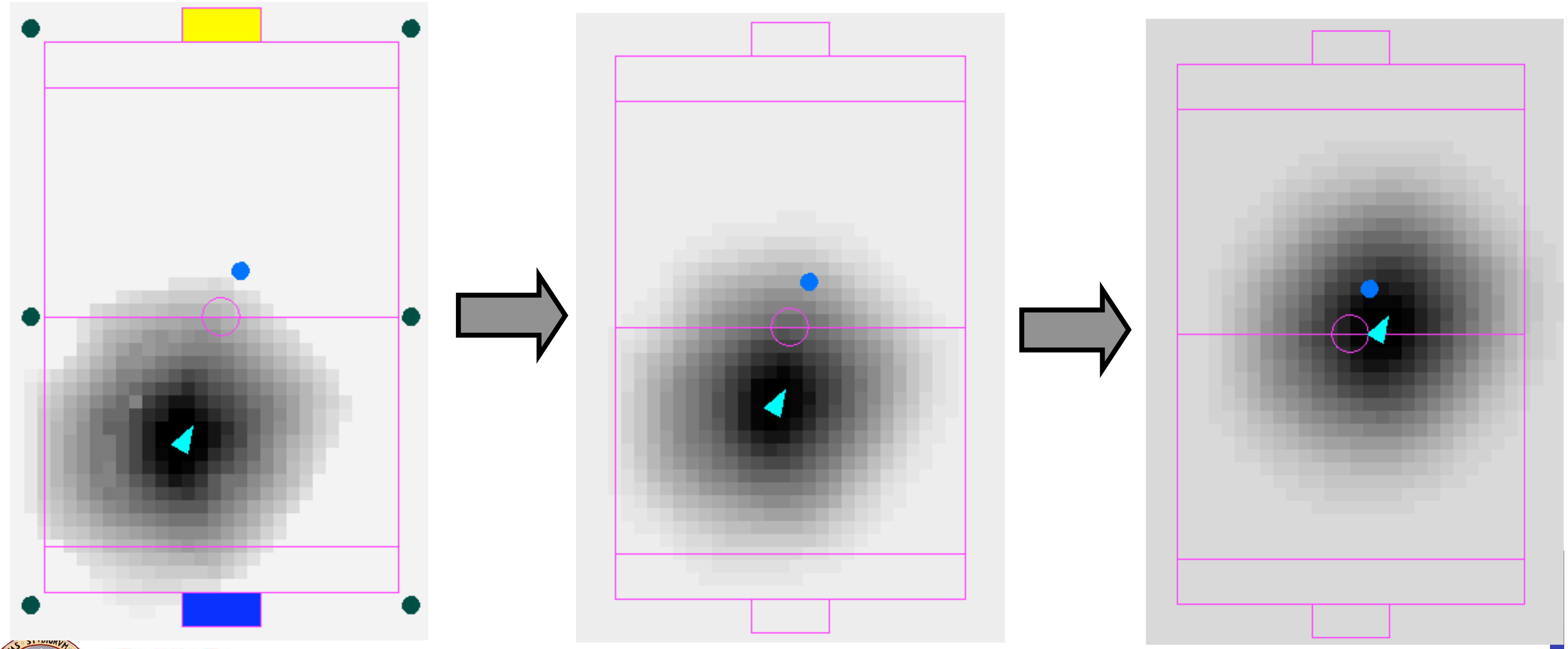




- Centre of gravity (defuzzification)
- Several measures of quality
  - Bounding box
  - Highest peak
  - Bias

Odometric Update

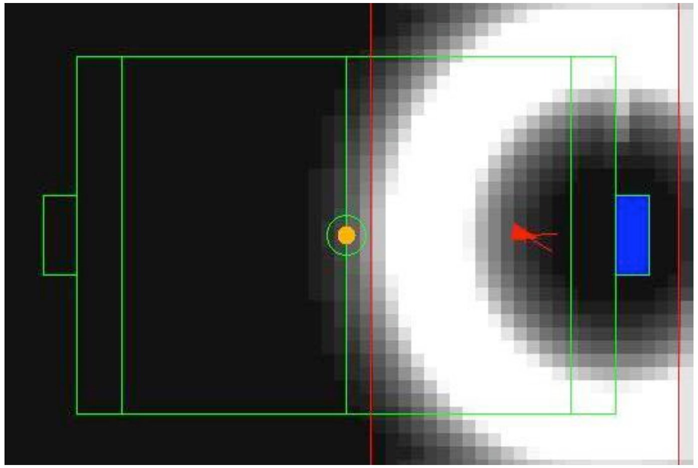
- When the robot moves
  - Translate the distribution (motion model)
  - Blur it (uncertainty model)



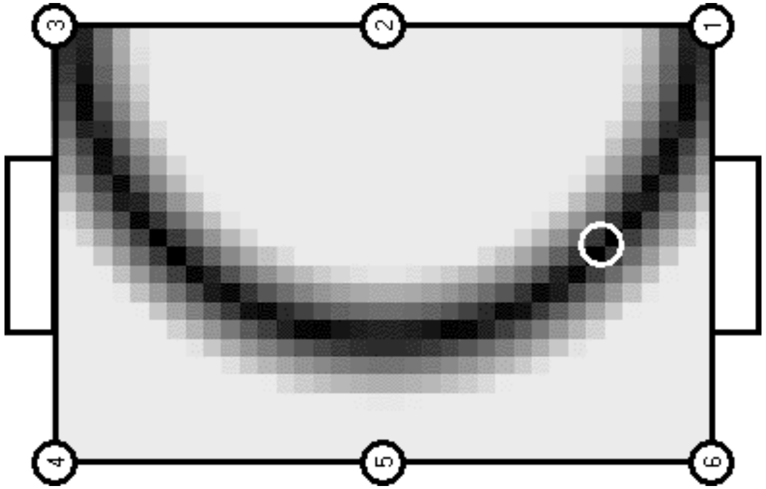
GM

Sensor Update

- When a landmark or net is detected
  - Build distribution of possible observation poses
  - Sensor model
    - Independent modelling of position and heading uncertainty

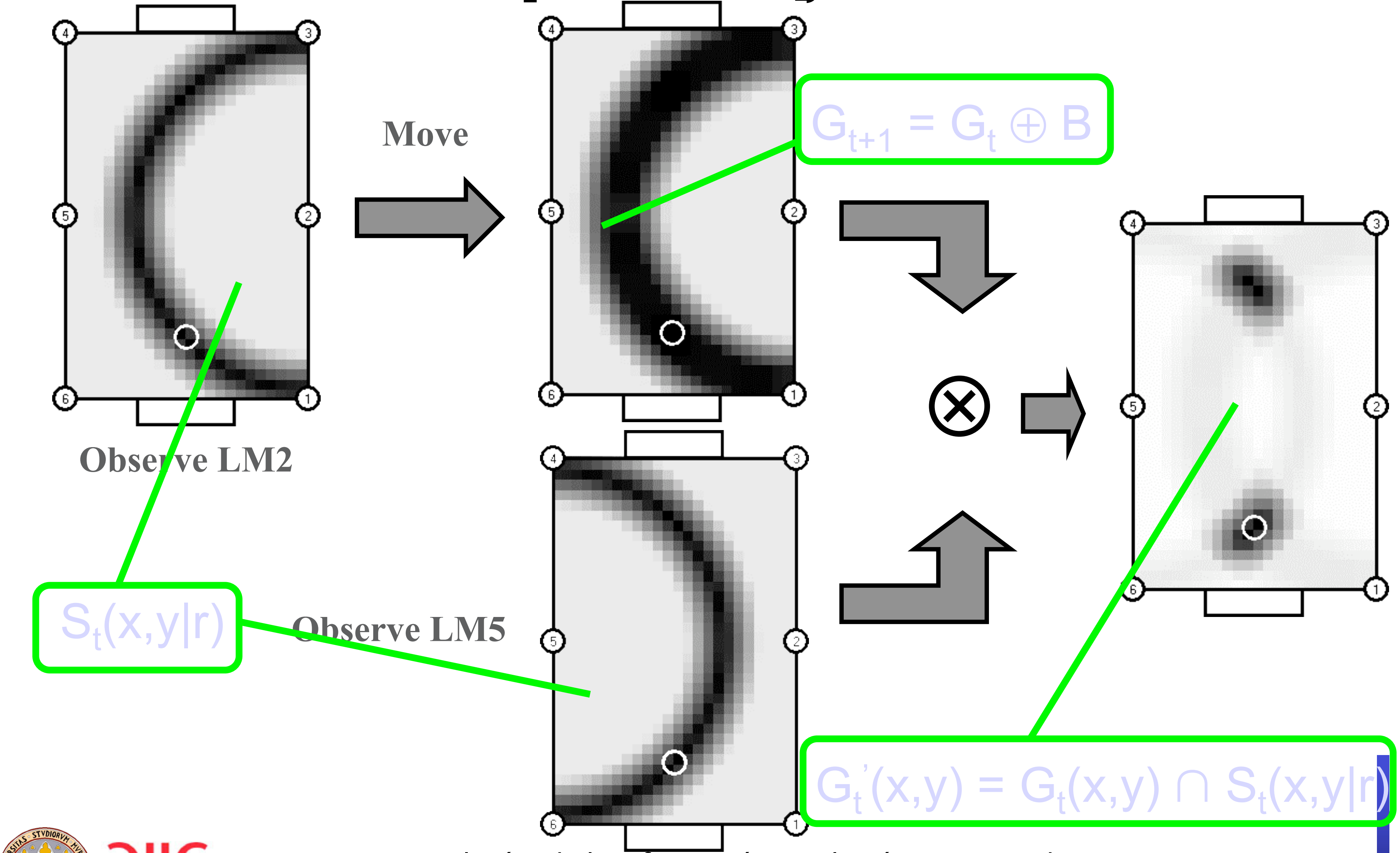


Blue Net



Middle Landmark

The Predict-Update Cycle



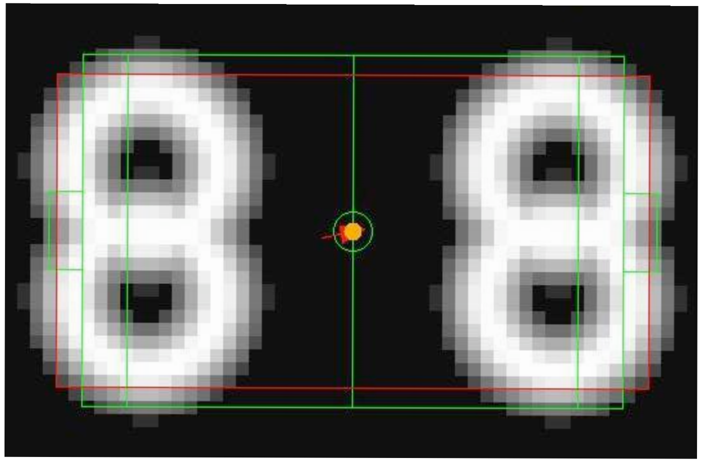
$G_{t+1} = G_t \oplus B$

$S_t(x,y|r)$

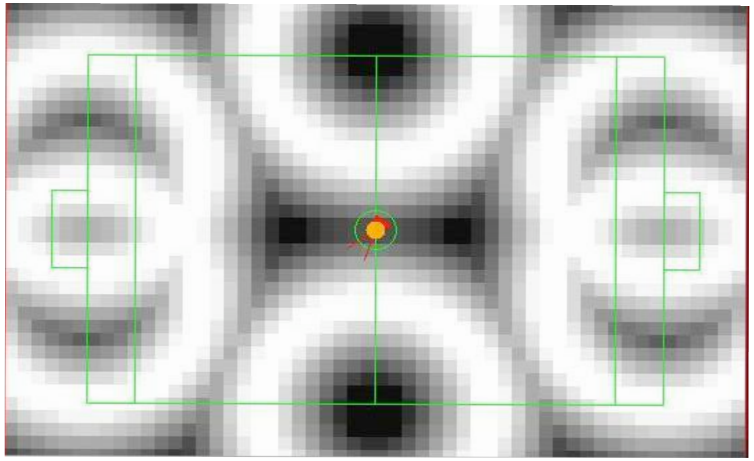
$G_t'(x,y) = G_t(x,y) \cap S_t(x,y|r)$

Extended Sensor Update

- When a feature is detected
  - – Build distribution of possible observation poses
  - – Each feature type has a different sensor model
  - – The features are not unique (ambiguity)



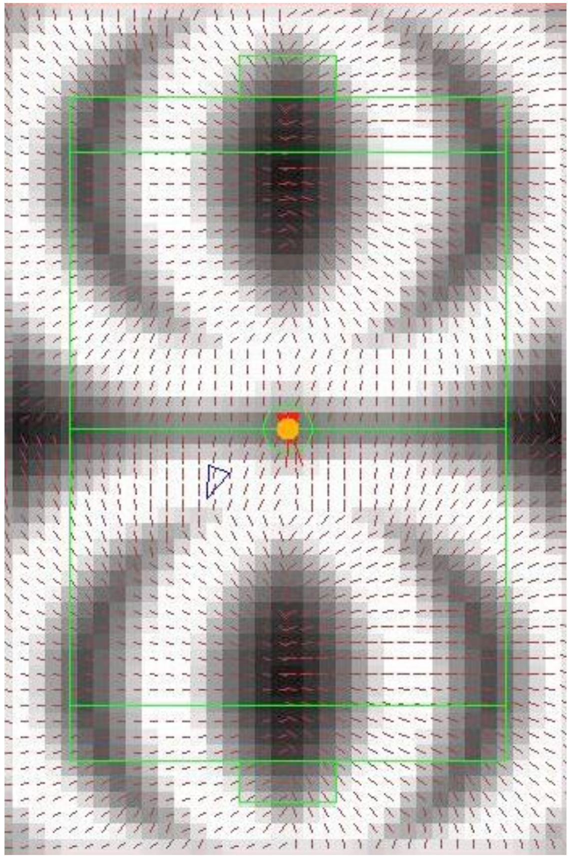
C field



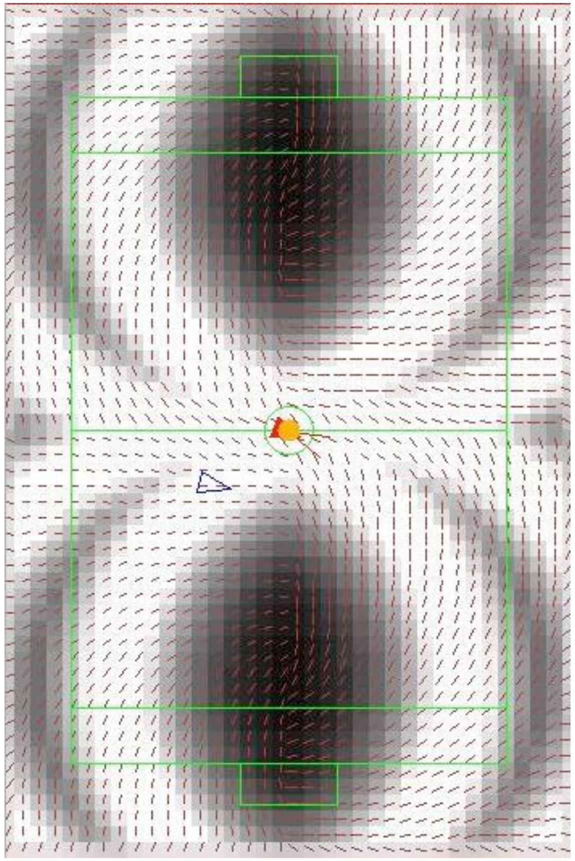
T field

GM

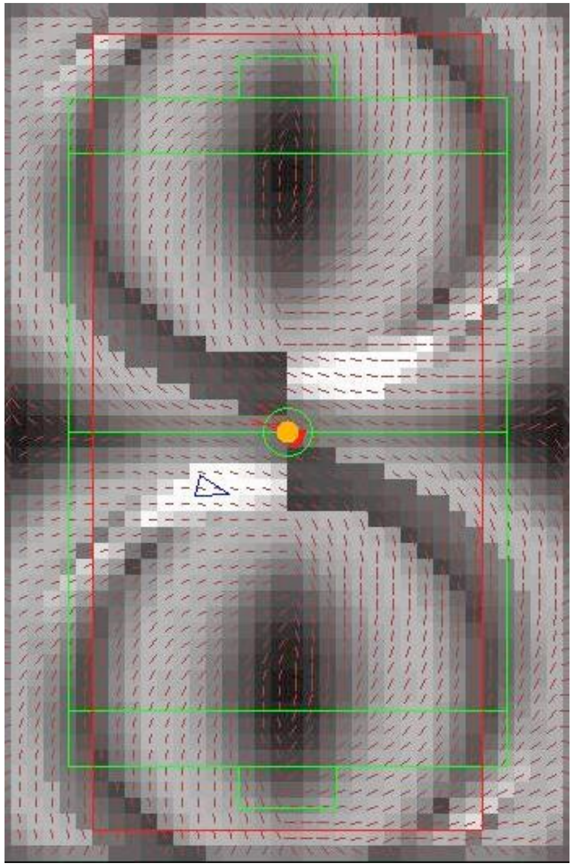
Examples (1)



Observe C

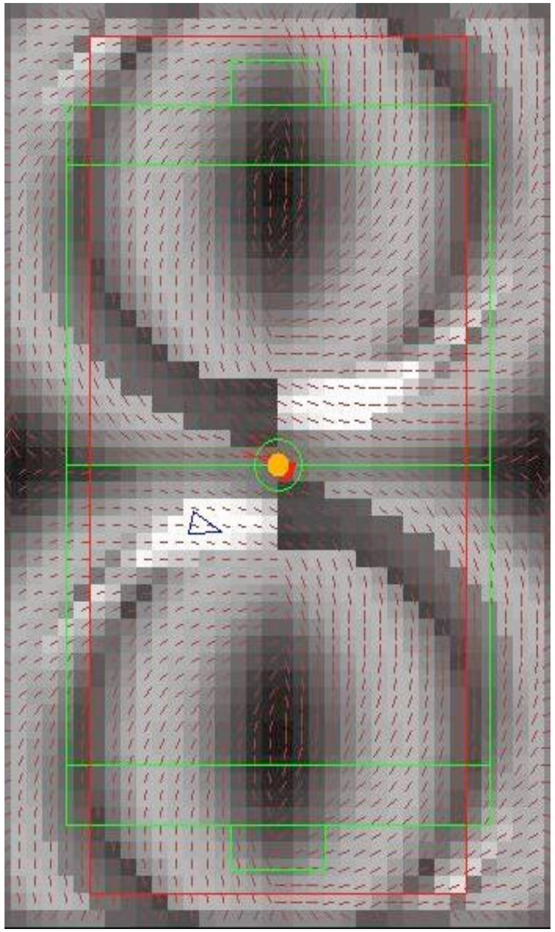


Move

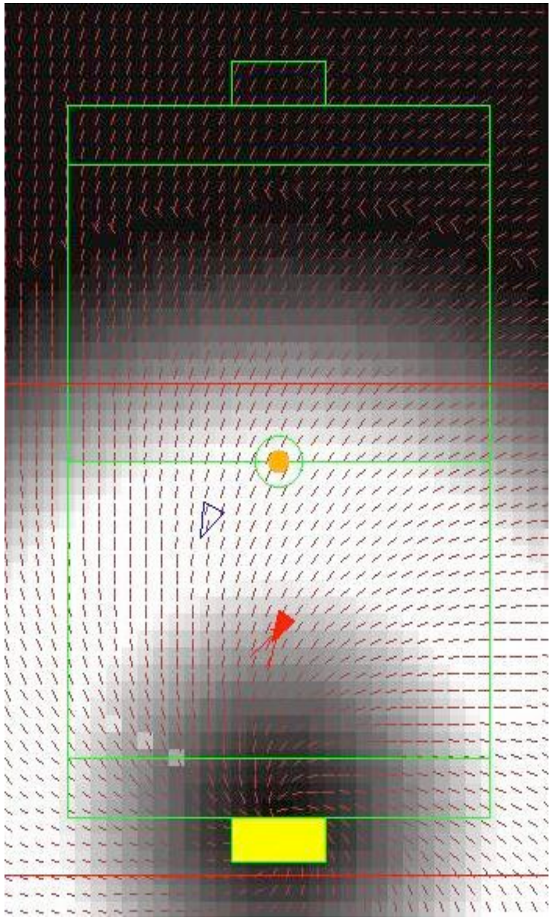


Fusion

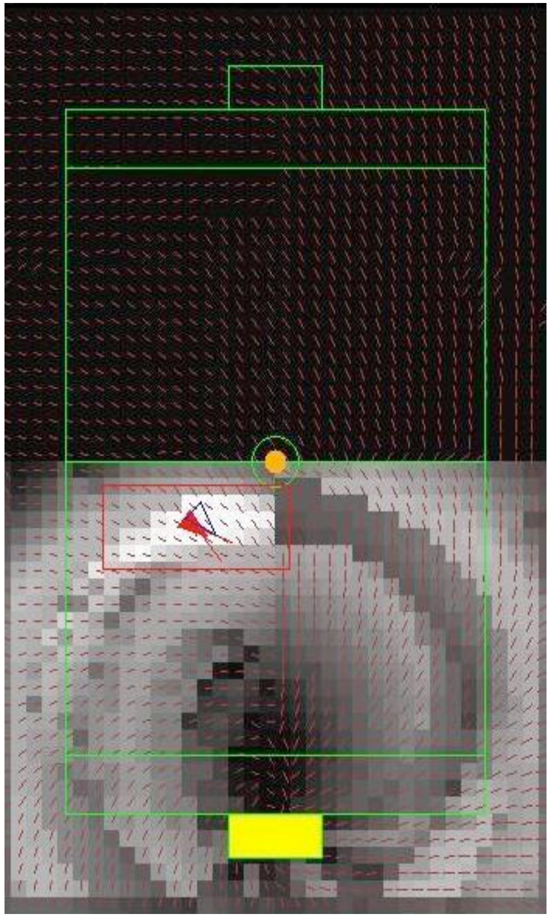
Examples (2)



Prev. State



Observe Net

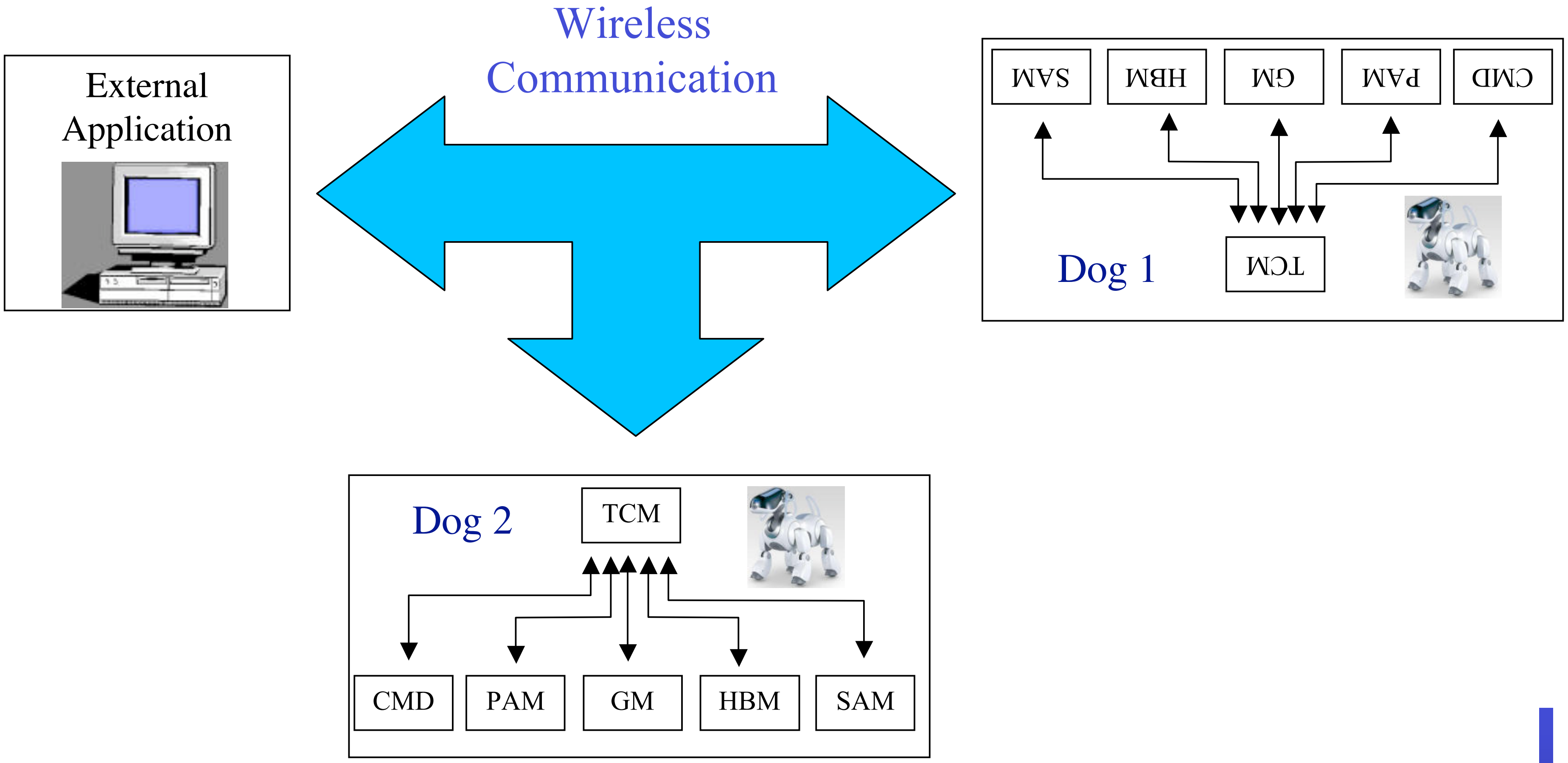


Fusion

Team Coordination Module

TCM

Inter-Module Communication


External Application
Wireless Communication
SAM
HBM
GM
PAM
CMD
Dog 1
TCM
Dog 2
TCM
CMD
PAM
GM
HBM
SAM