



Pre-production

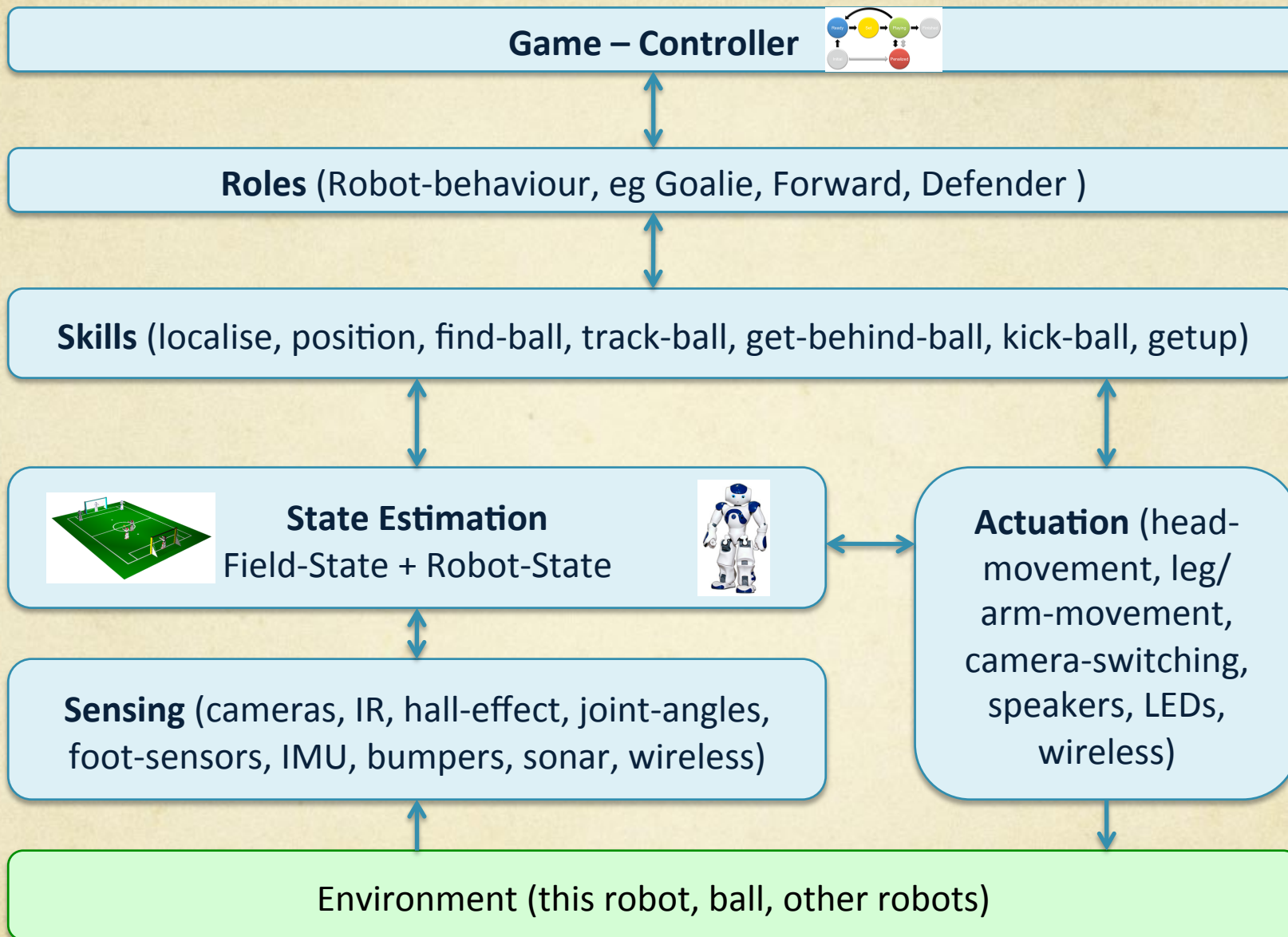
# Robocup 2010 Version 1 Storyboard Prototype

13<sup>th</sup> October 2009

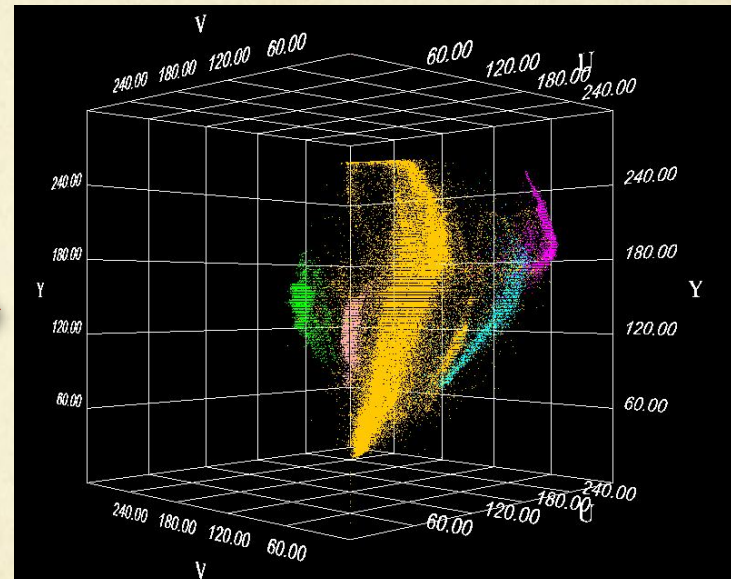
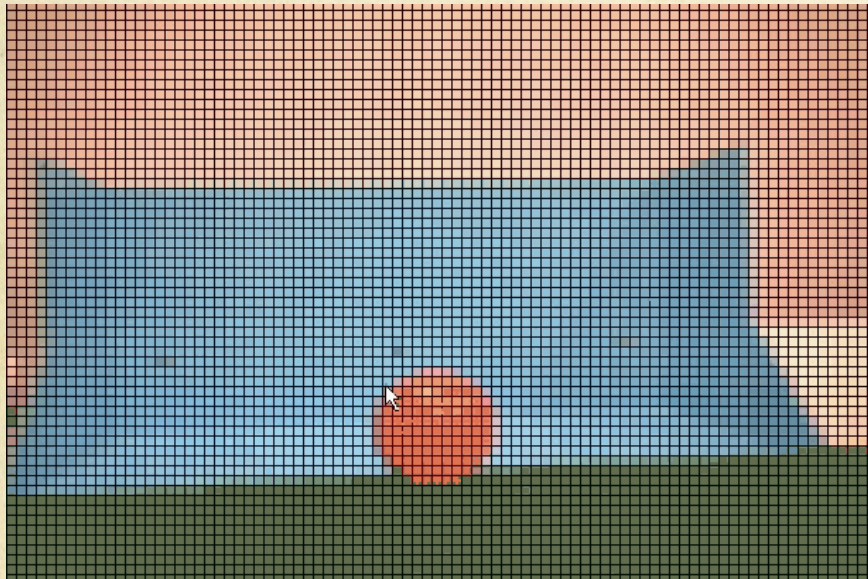
Bernhard Hengst

# Developmental Research Strategy

- Fail-fast, fail cheap
- We should accept poor performance but insist it is complete
- Version 1 objective is to have a single Nao kick a ball into the goal
- While we develop and test version 1, we store up ideas for version 2, 3 , 4 ....



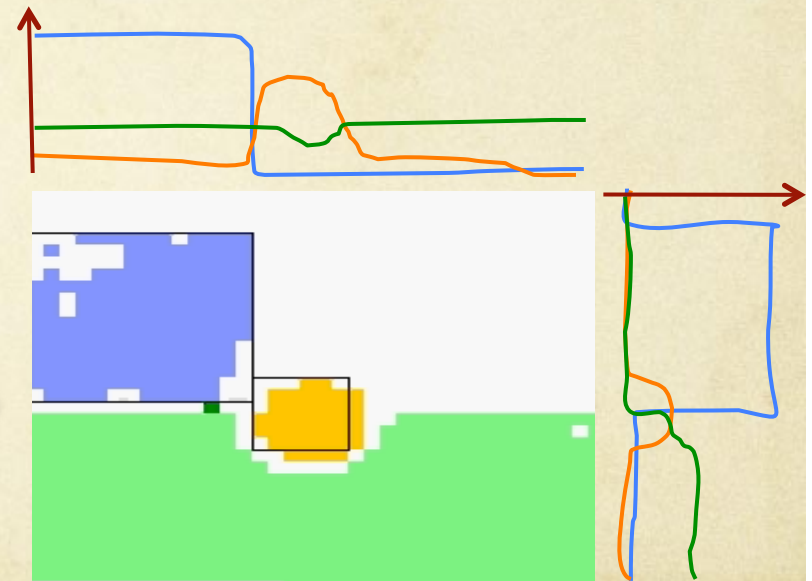
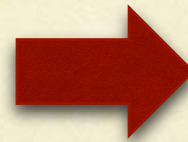
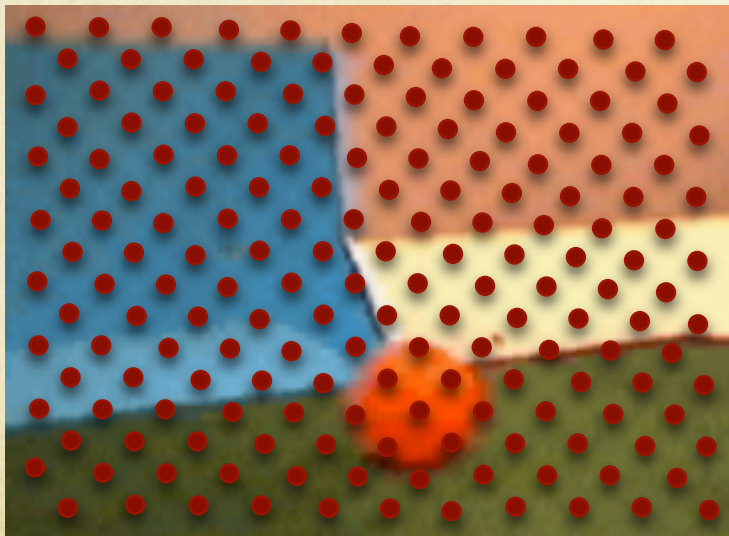
# Colour Classification



Off-Nao Demonstration

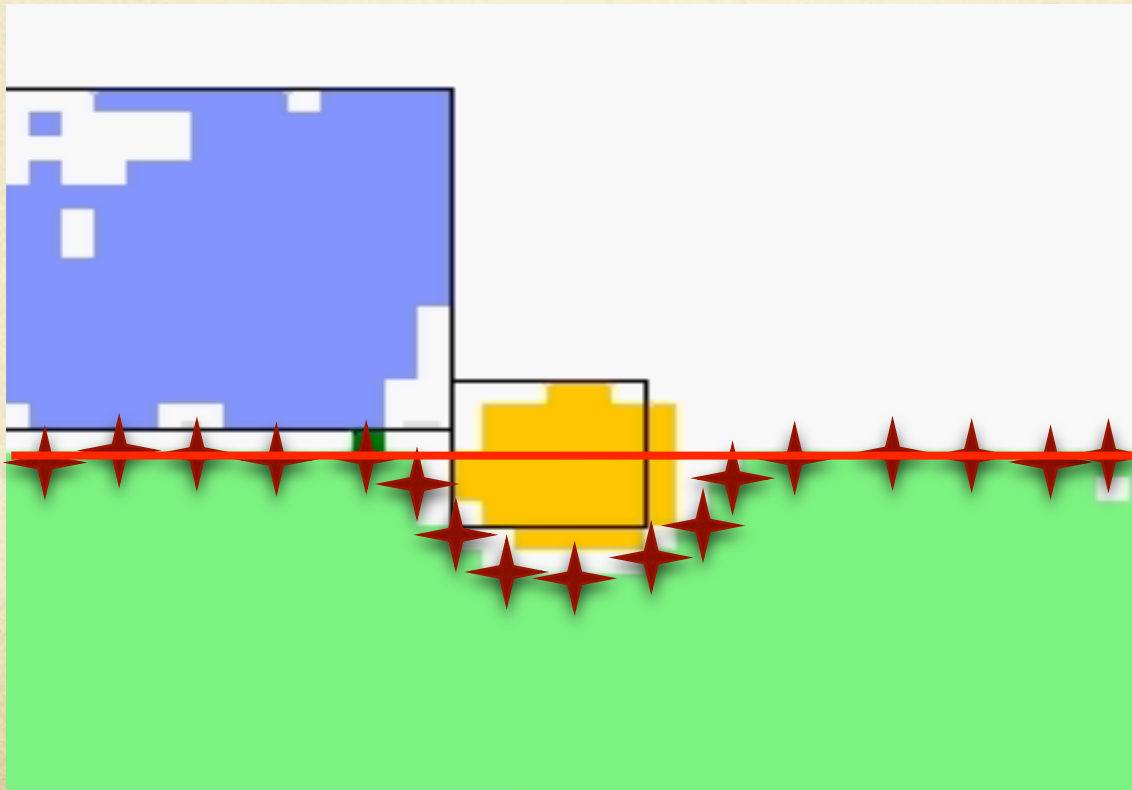
# Vision Saliency Preprocess

- Subsample and classify 640 x 480 image down to 160 x 120 (80 x 60?)
- By processing every 4<sup>th</sup> (8<sup>th</sup>?) pixel in the image, both vert. & horiz.
- This gives us information about the field edge, ball and goal locations



# Field-edge Location

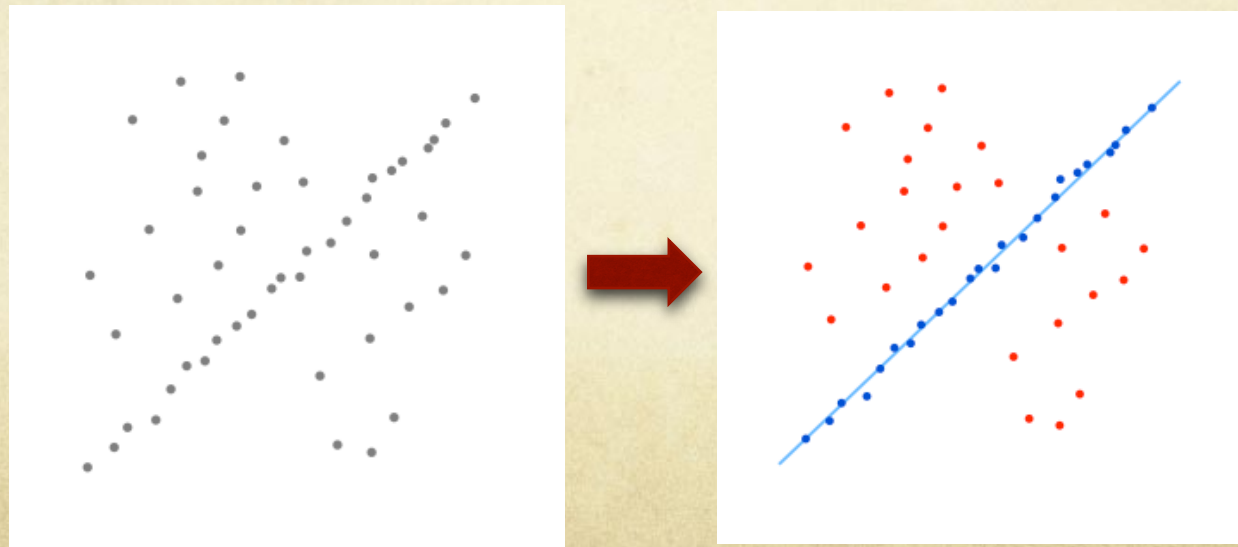
- Find green to non-green transition points
- Find field-edge lines using RANSAC



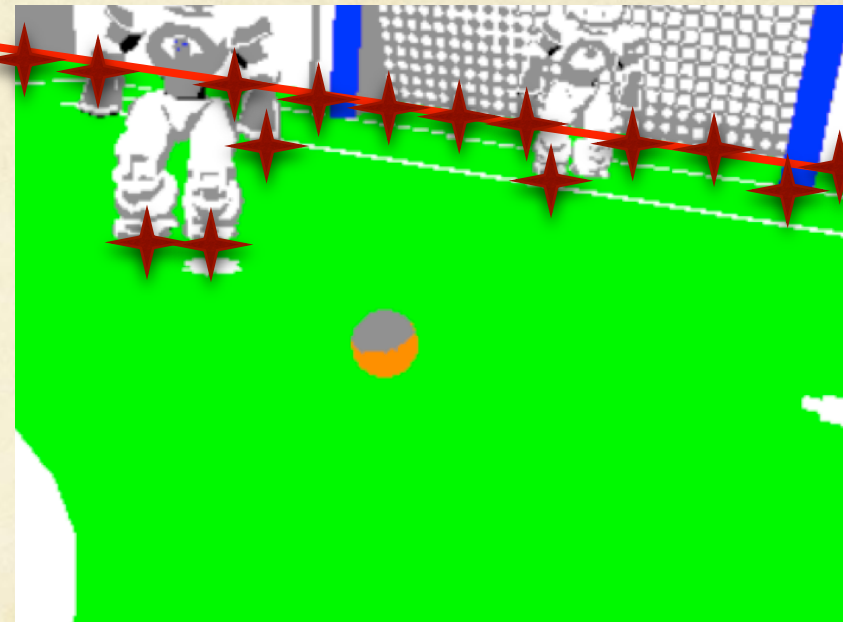
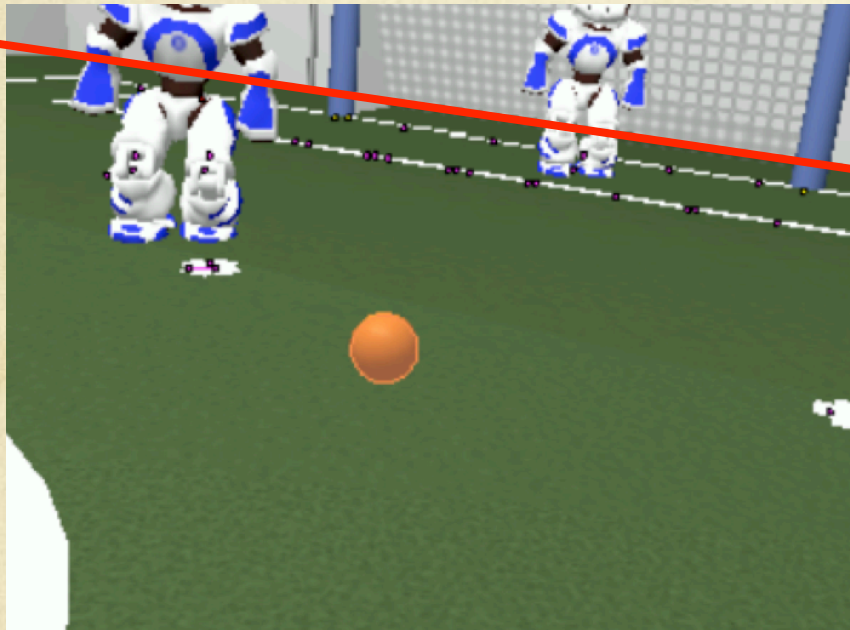
ASIDE

# RANSAC

- RANSAC is an abbreviation for "RANdom SAMple Consensus"
- It is an iterative method to estimate parameters of a mathematical model from a set of observed data which contains outliers.



# Field-edge display Off-Nao

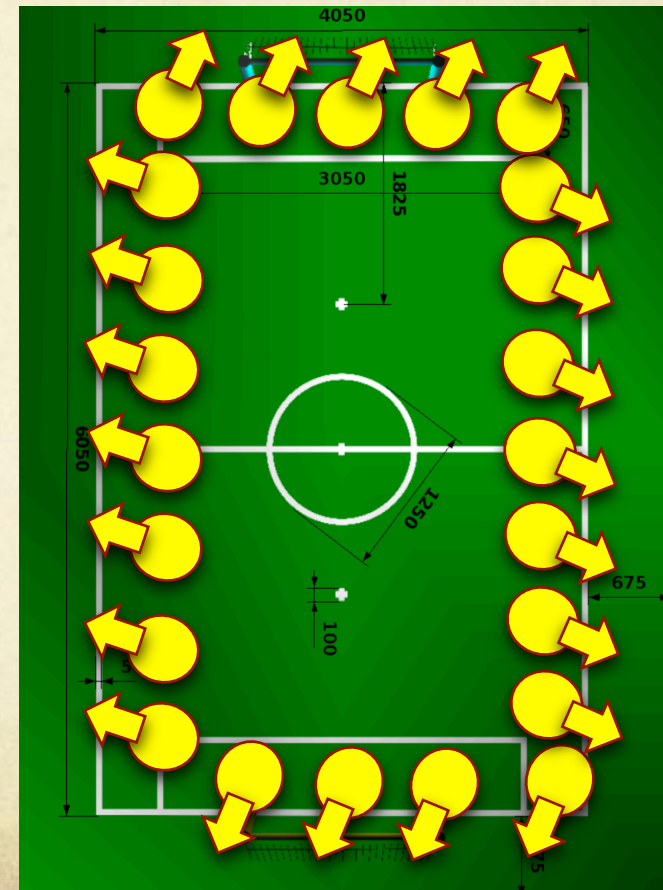
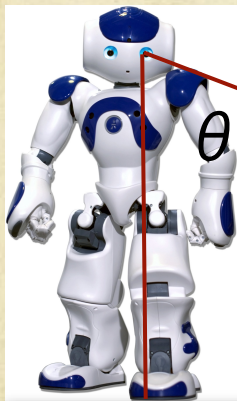




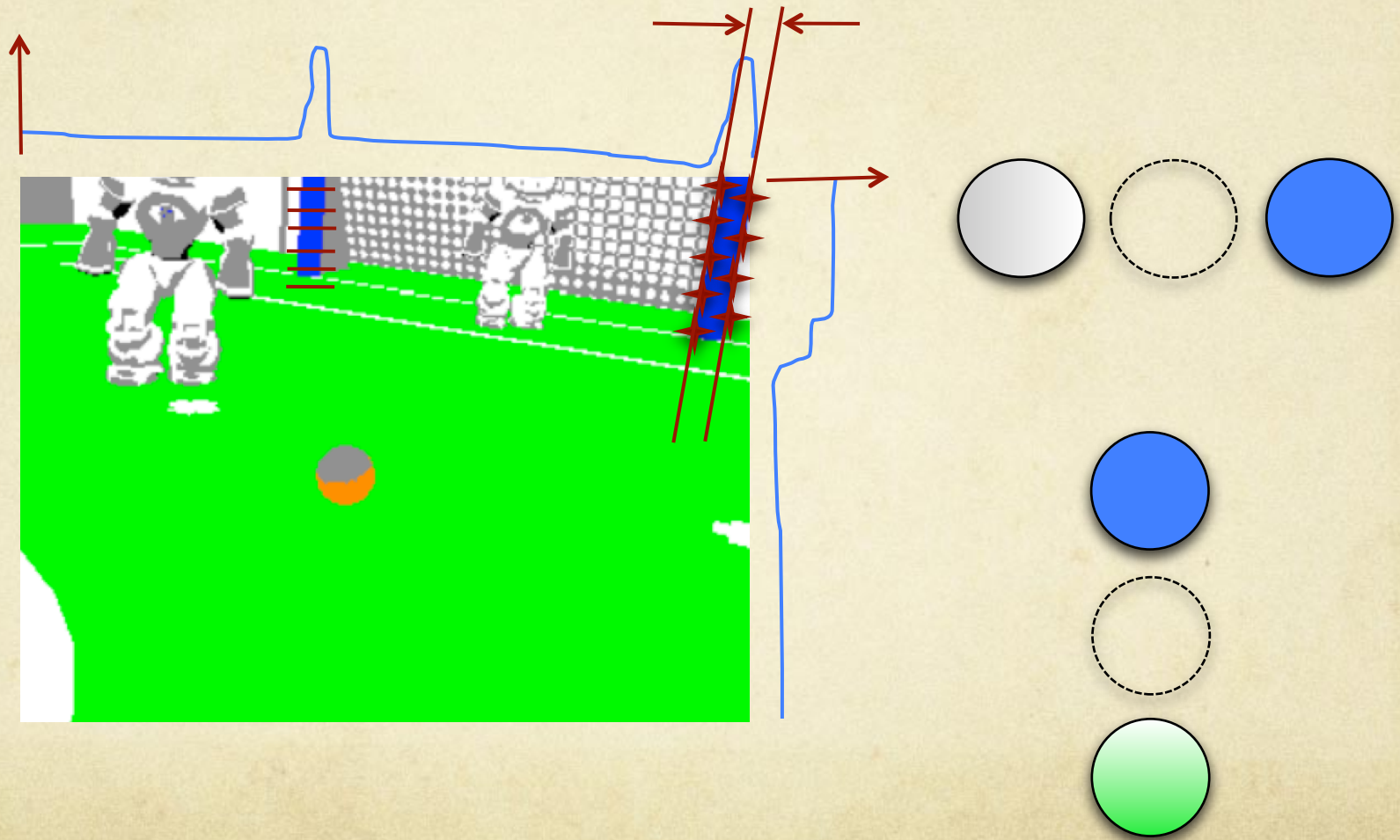
ASIDE

# Field-edge Position Observations

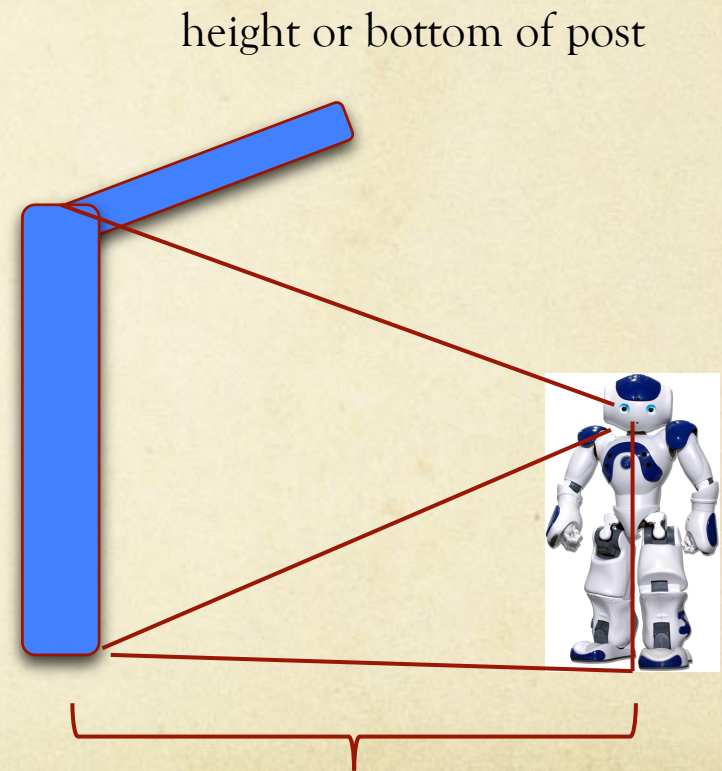
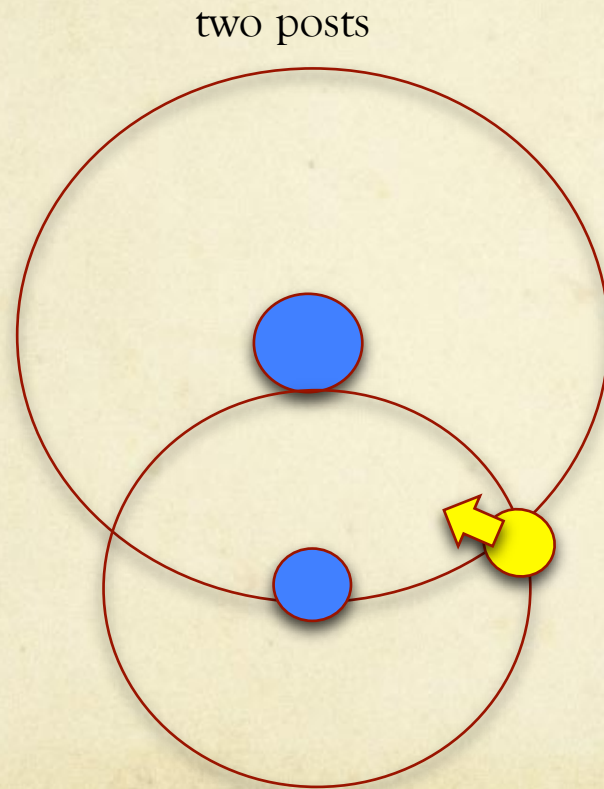
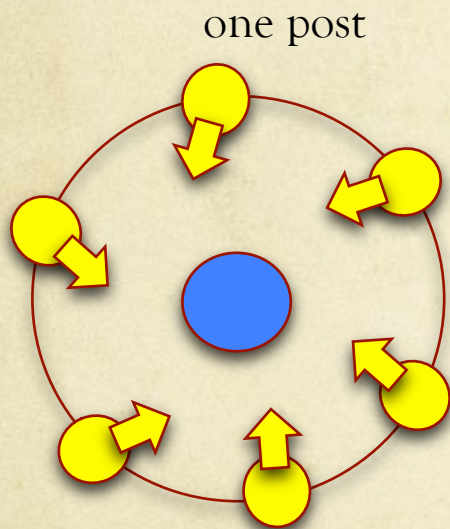
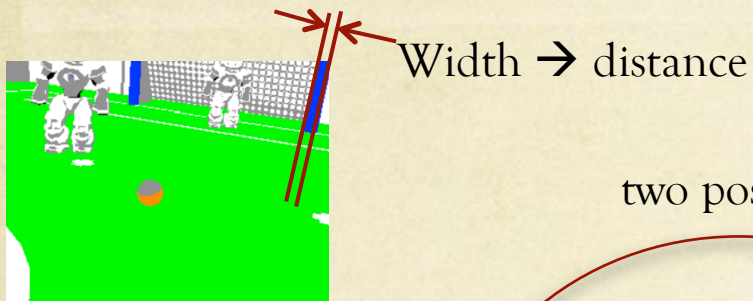
- Project vision field-edge into plan view
- Find likely positions of robot
  - One line determines rectangle
  - Two lines determine 4 points



# Goal-post Identification



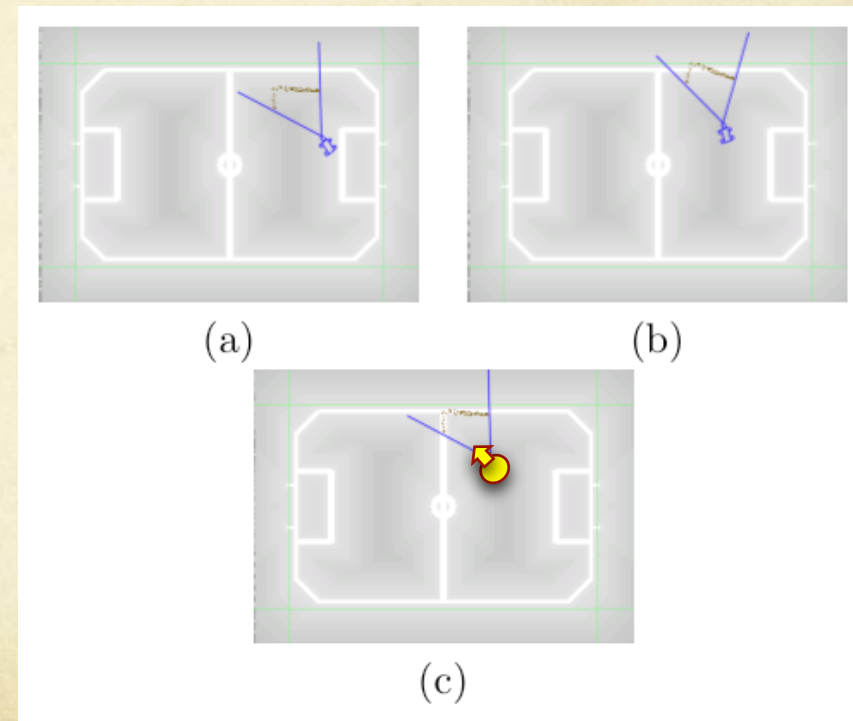
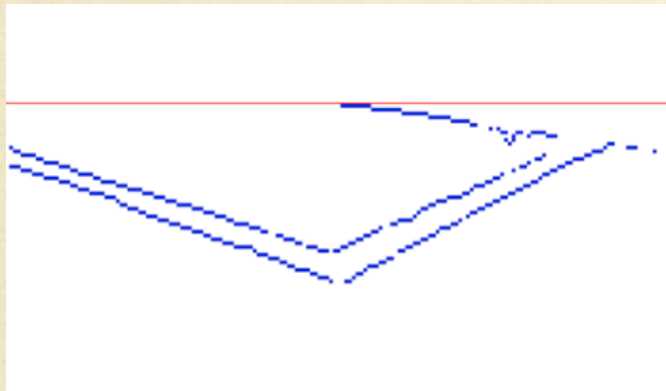
# Position Estimate from Goal Posts



ASIDE

# Position Estimate from Field-lines

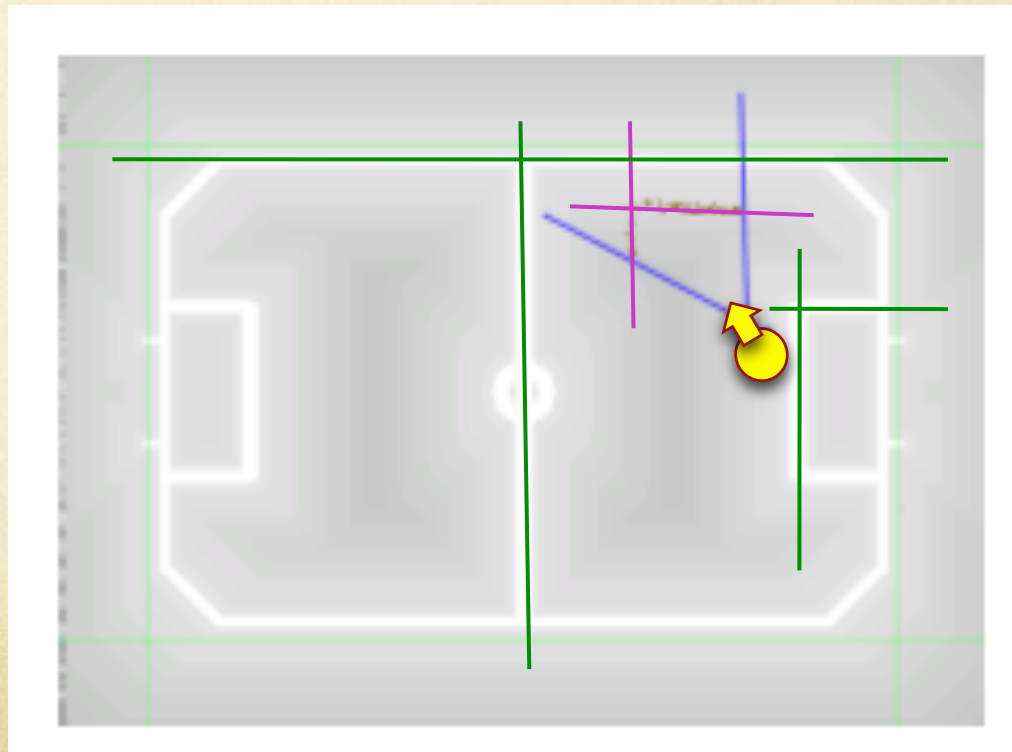
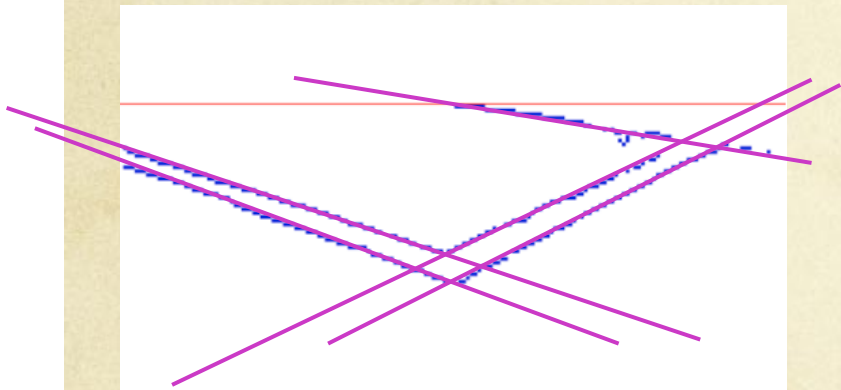
## Matching Points



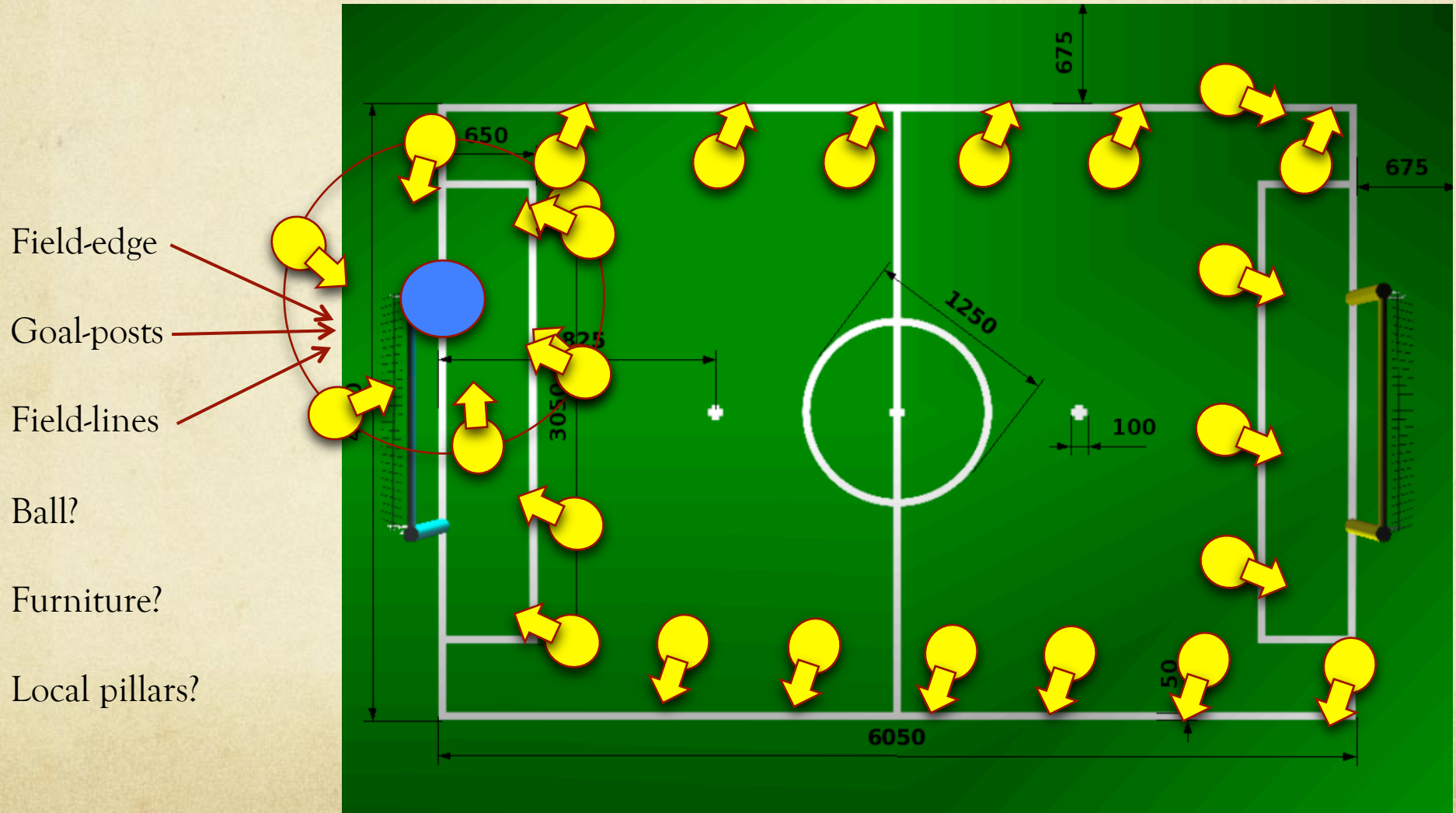
ASIDE

# Position Estimate from Field-lines

## Matching Lines



# Robot Localisation

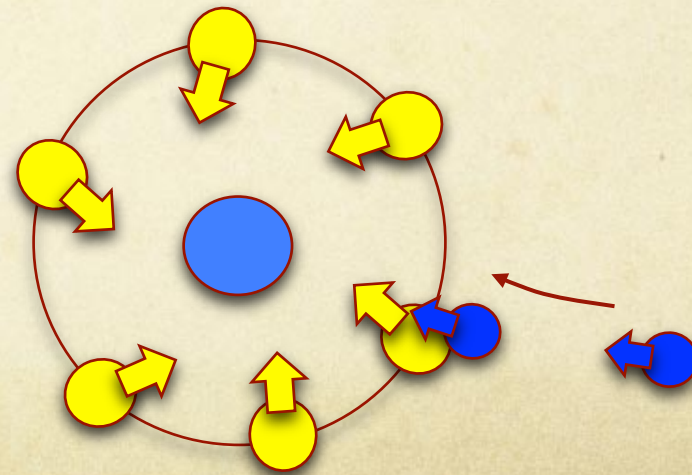
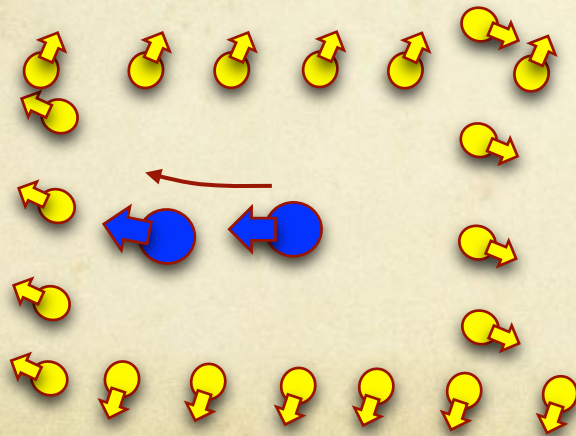


# Bayesian Filter Considerations

- Kalman filter and variations. Fast, but poor approximation to multi-modal distributions generated by SPL observations
- Particle filters. Represent multi-modal distributions directly, but suffer from heavy computational load with many particles
- Variables to model ie robot-position, ball-position, other robots (own team, opposition team)
- Could use multiple-model KF (Uther and Quinlan/Middleton), possibly with less variables eg 3 for robot + 2 (or 4) for ball, or start with PF and switch to KF
- But, for version 1, start with simpler heuristic

# Localisation Heuristic (Version 1)

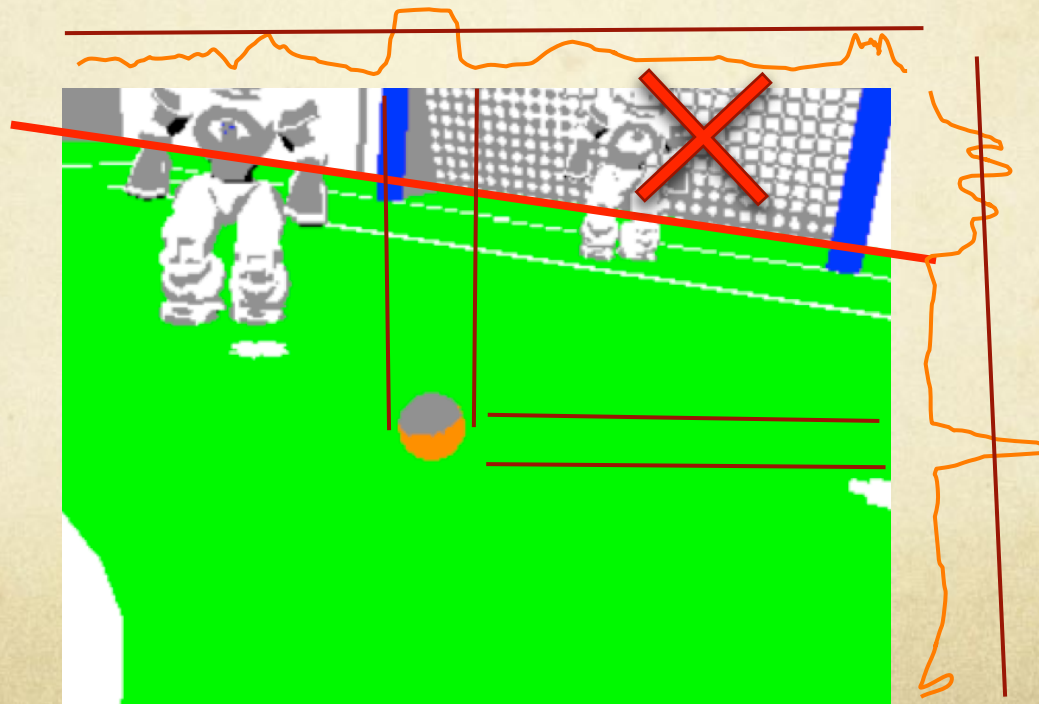
- Find closest distance ( $\Delta x$ ,  $\Delta y$ ,  $\Delta \theta$ ) from current position to observation.
- Update robot position towards observed position based on relative confidence between current and observed
- Off-Nao Demonstration





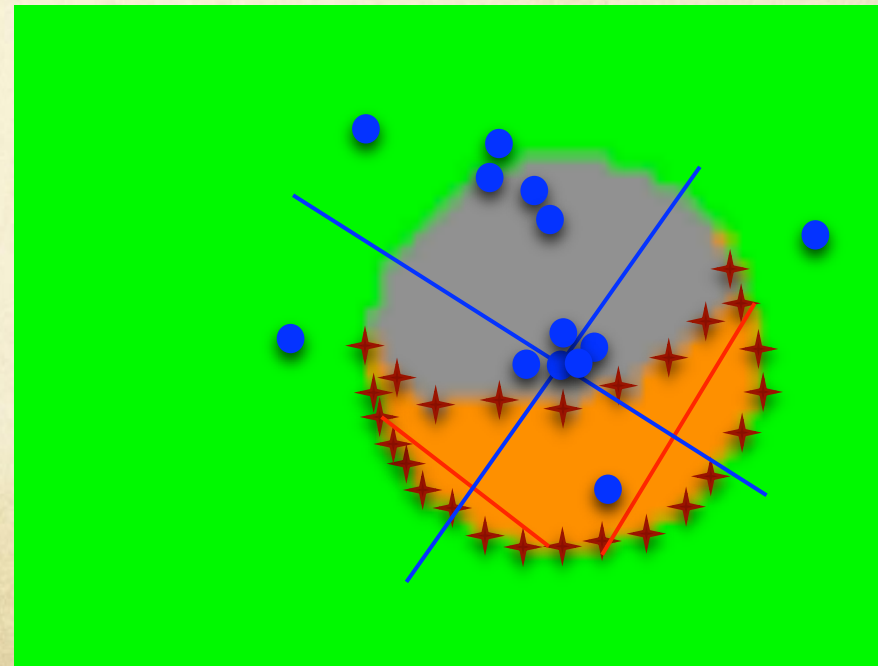
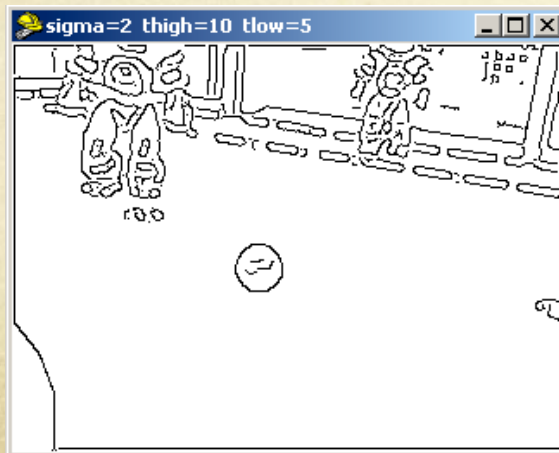
# Ball Position (1)

- In 160 x 120 saliency image look for likely ball places and size (eg max orange pixel tally both horizontally and vertically)
- Ignore any positions above field-edge line in image



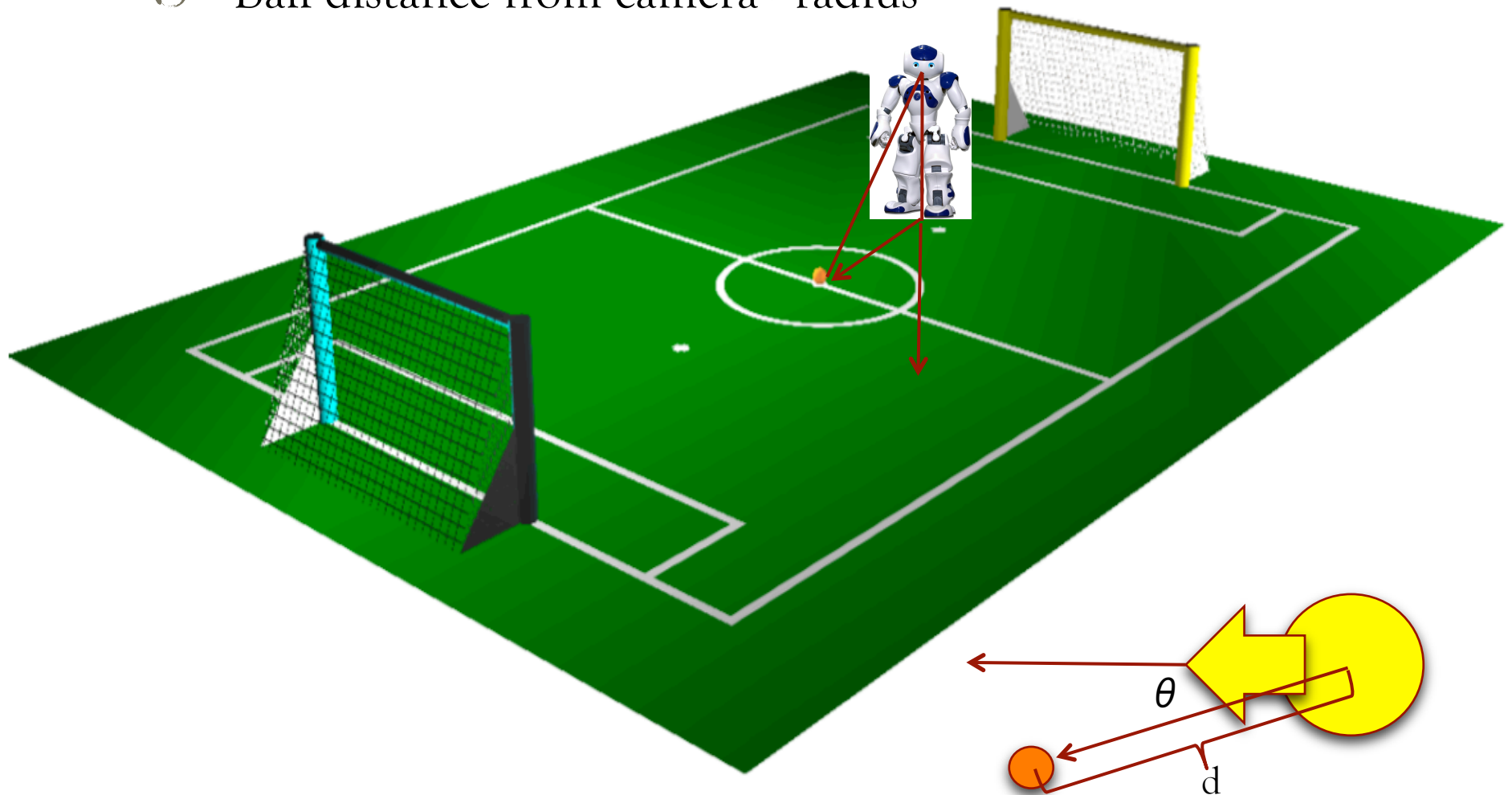
# Ball Position (2)

- Extract 160 x 120 sub-image (fovea) from 640 x 480 image to fit ball [or locate and track ball boundary directly - Carl's idea]
- Find ball-edge pixels ie orange to non-orange transitions
- Use median for observation measure of center and hence radius.
  - Ref Alex North report

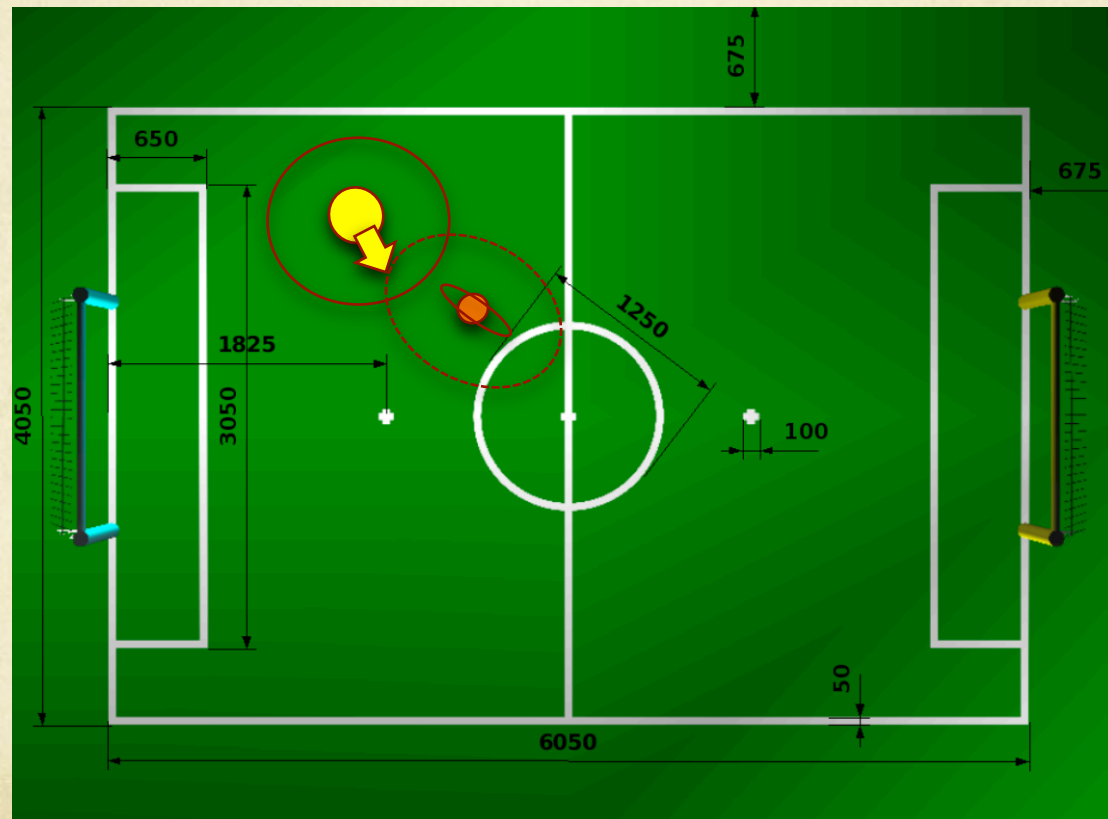


# Ball Position (3)

- Ball distance from camera - radius



# Off-Nao Robot-position & Ball



# Localise Skill

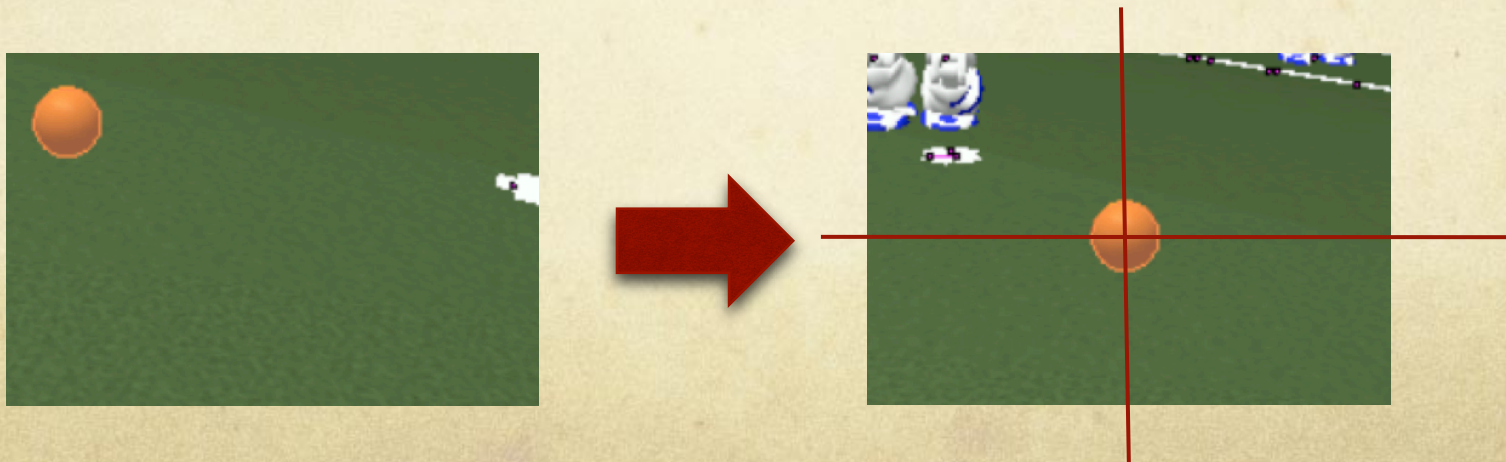
- Localise first (before looking for ball)
- Use robot position variance (confidence, uncertainty) as a threshold parameter
- Look around for goals [later field-edges, field-lines, ball, natural landmarks]
- Off-Nao Deonstration

# Find-Ball Skill

- Like a survivor search mission at sea
- Need to determine a good search heuristics
- Criteria - use any prior knowledge (ball location probability is not uniform)
  - Location on the field (can see more of the field in certain locations)
  - Search below field-edge
  - Minimise walking/head movements
  - Last known position of ball and which direction it was rolling
  - Minimise inadvertent ball pushing in wrong direction
  - Don't forget - ball may be occluded
  - Hints from other robots ..... etc, etc.
  - Start with simple heuristic for Verison 1
- Off-Nao Demonstration

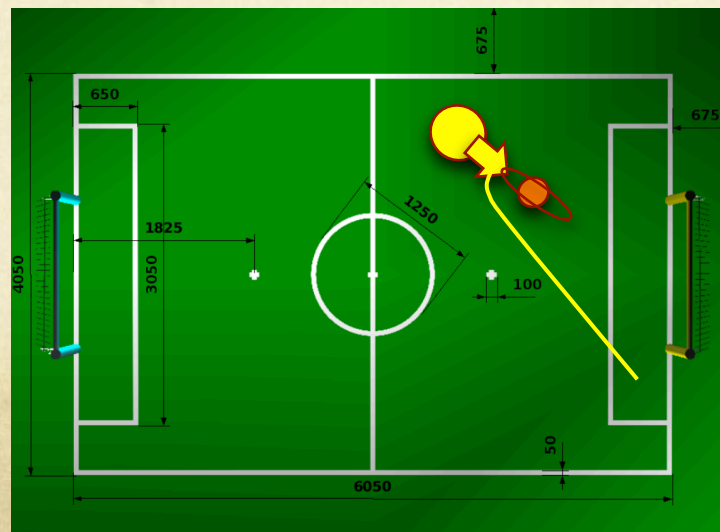
# Ball Tracking Skill

- Software saccade to keep fovea on ball in 640 x 480 image, with ball centered in fovea
- Move head smoothly (and switch cameras) to keep fovea in center of image
- Off-Nao Demonstration



# Walk-behind Ball Skill

- Use Aldebran walk (V1) – need to control movements
  - forward – backward
  - left-right
  - turn CW and CCW
- Walk to ball skill ie correct reactively
- Plan shortest path to line up ball and target-goal



Demonstration



# Kick-Ball Skill

- Walk into ball (V1)
- Tap ball in direction of Goal
- Sideways tap

# Kick Goal Behaviour (V1)

- Repeat until goal scored
  - if ( robot-position uncertain) Localise
  - else if ( relative ball-position uncertain) Find-Ball
  - else if (ball not lined-up) Track-Ball & Walk-behind Ball
  - else Kick-Ball
  
- Note: Vision, Robot/Ball positioning and movement tasks are running in separate higher frequency threads