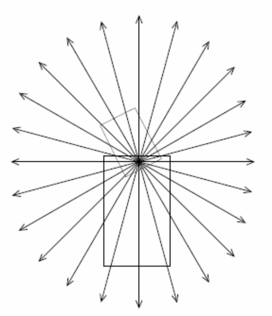
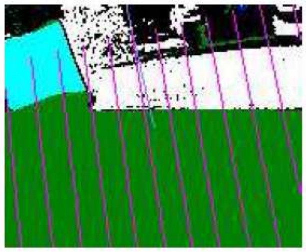
Robocup Team Reviews
Carnegie Mellon University





CSE398/498
07 Feb 06

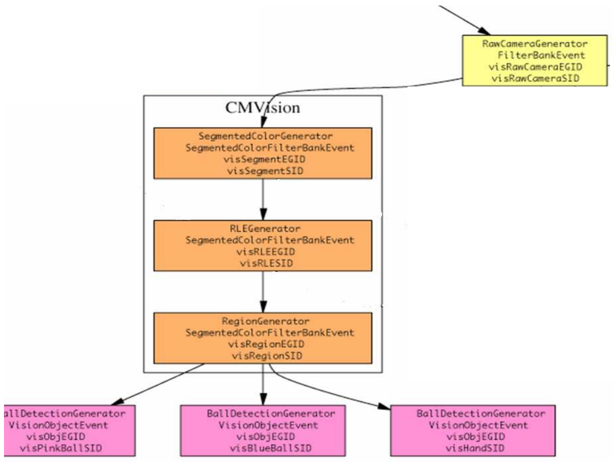
References

- *"CMPack-02: CMU's Legged Robot Soccer Team,"* M. Veloso *et al*
- *"Visual Sonar: Fast Obstacle Avoidance Using Monocular Vision,"* S. Lenser and M. Veloso, IROS 2003, Las Vegas, USA

What are Events?

- Objects of course
- Consist of three parts
    1. Generator (e.g. button generator, vision system, timer)
    2. Source (e.g. which button?)
    3. Type (EventBase::activateETID, EventBase::deactivateETID, EventBase::statusETID)
- These enable the listening behavior to shape its actions

CMU High Level Vision: CMVision



http://www-2.cs.cmu.edu/~tekkotsu/Vision.html

High Level Vision: Ball Detection

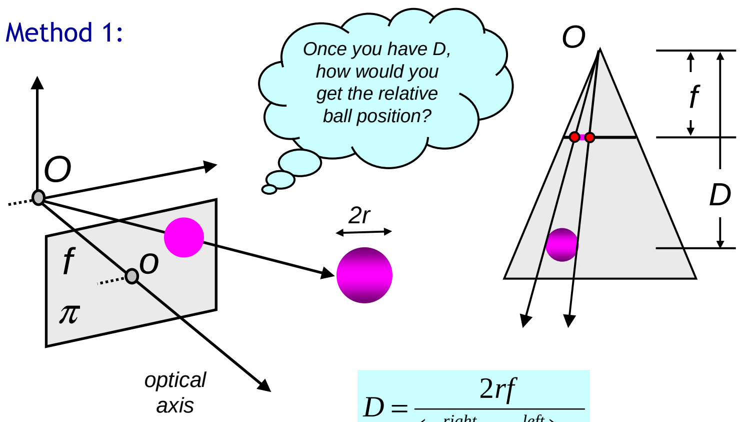
- Each ball detection generator receives a list of the 10 largest (in this case) pink regions detected in the image
- The region with the greatest “confidence level” is returned as a ball (maybe)
- Confidence threshold specified in tekkotsu.cfg

Ball Detection Confidence Calculation

- Constraint filters:
  - Tilt angle (<5º above robot head)
  - Region size (height: 3 pixels, width: 3 pixels, area: 7 pixels)
  - Fringe effects (red uniform on yellow goal)
  - Proximity constraints
- Real value filters:
  - Aspect ratio of the bounding box (BB)
  - Pixel area vs. BB area
  - Divergence in angle between position calculations
- Confidence calculated from the product of real value filters
- Confidence scaled based upon the number of pixels to prevent rejection of very close balls

Ball Location Calculation

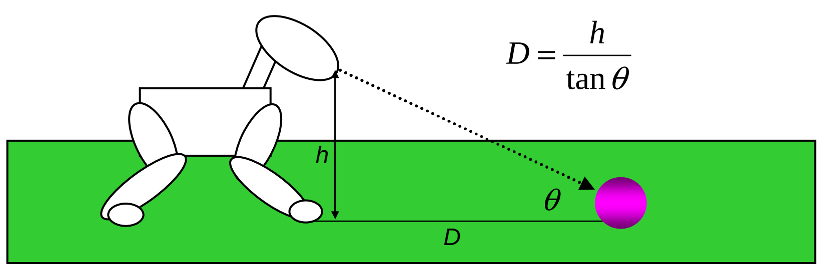
- Two methods used to estimate the relative position of the ball
- Both rely on using coordinate transformations to determine the position of the camera/ball
- Method 1:



$$D = \frac{2rf}{(x_{im}^{right} - x_{im}^{left})s_x}$$

Ball Location Calculation

- Two methods used to estimate the relative position of the ball
- Both rely on using coordinate transformations to determine the position of the camera/ball
- Method 2:

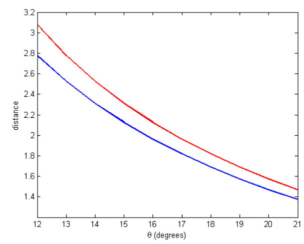


$$D = \frac{h}{\tan\theta}$$

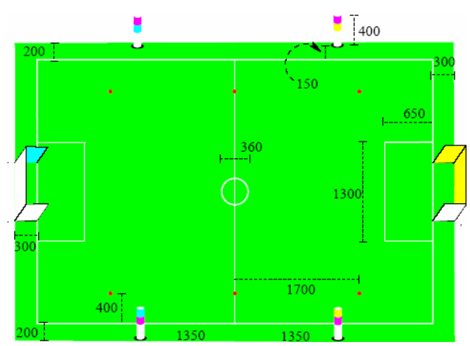
Location Method Comparisons

- Method 1:
  - Advantage: Less sensitive to errors in camera orientation
  - Disadvantage: Assumes that the entire ball is visible
- Method 2:
  - Advantage: Less sensitive to ball occlusions or when the ball is at the edge of the picture frame
  - Disadvantage: Noise can cause significant errors at longer distances

$$D = \frac{h}{\tan\theta} \quad \Rightarrow \frac{\partial D}{\partial \theta} = \frac{-h}{\sin^2\theta}$$



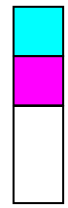
High Level Vision: Marker Detection



- Marker detection will (probably) be your basis for robot localization as these serve as landmarks for position estimation
- Need to correctly associate pairs of segmented regions with the correct landmarks

** www.robocup.org*

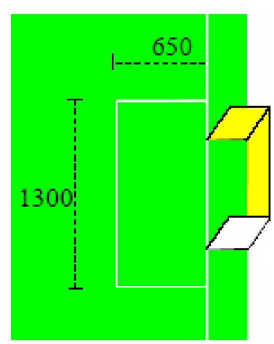
Marker Detection & Confidence Calculation



- Detection:
  - Consider all pairs of pink and yellow/cyan out of the 10 largest regions of each color
  - The most confident marker readings are used for localization
- Real value filters for confidence calculation:
  - Angular separation of the two regions
  - Relative size of the marker region pair (they should be roughly equal)
  - Relative size of each region relative to the square of the distance between them
- Location Calculation:
  - Take the average of the rays to each region to form the landmark plane
  - Use *a priori* knowledge of landmark size to estimate position

High Level Vision: Goal Detection

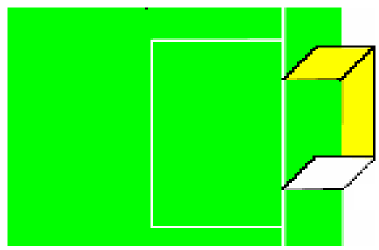
- Goal detection based upon corner detection to estimate its position
- Goal is represented as three objects:
  - Left edge
  - Right edge
  - Central object
- The central object is for behaviors requiring rough estimates
- The edges are for localization and aiming



* www.robocup.org

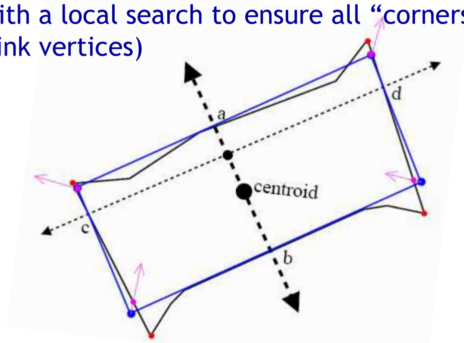
Central Goal Object

- Relies on the aspect ratio of the goal and the goal's proximity to the field
- Angle to the goal is calculated from the centroid
- Distance to the goal is estimated using its height
- The bearing angle subtended by the goal is also calculated



Corner Detection for Edges

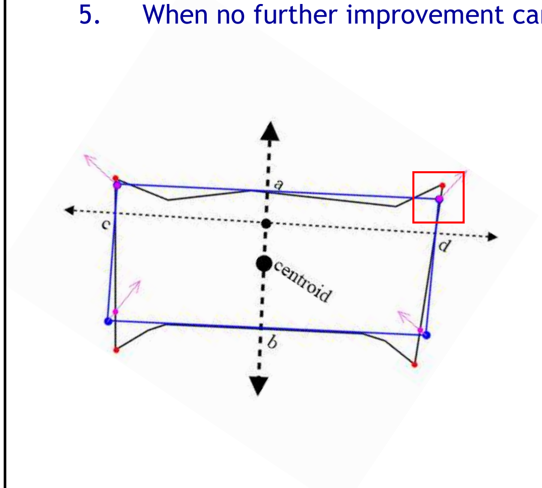
1. Start at the centroid of the goal region, and proceed in the +Z direction (world frame) until no more goal pixels are found (point a)
2. From the midpoint (no occlusion) between point a and the centroid, search in the +/- Y directions (again world frame) to form a "bounding rectangle" (blue box)
3. Refine with a local search to ensure all "corners" are goal pixels (pink vertices)

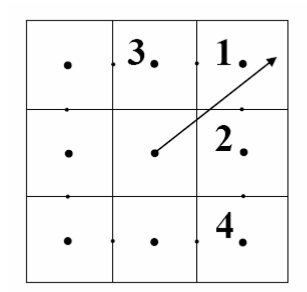


http://www-2.cs.cmu.edu/~coral-downloads/legged/papers/cmpack_2002_teamdesc.pdf

Corner Detection (cont'd)

4. Refine each corner estimate using a "gradient ascent" approach
5. When no further improvement can be achieved, terminate





http://www-2.cs.cmu.edu/~coral-downloads/legged/papers/cmpack_2002_teamdesc.pdf

Goal Edge Confidence Calculation

- Constraint filters:
  - Width & Height of the goal
  - Goal aspect ratio
  - Proximity to the field (green pixels beneath the goal)
- Real value filters:
  - Goal orientation
- Location Calculation:
  - Use the height of the goal and the corners to estimate the distance to the goal

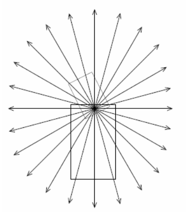
Robot Detection

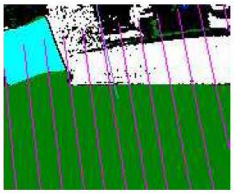
- Relies upon color segmentation

- Need to associate multiple regions from multiple dogs to multiple regions on individual dogs
- Again, this is based primarily on heuristics:
  - Keep regions separate when there is a lot of green between them or when they are far apart

Visual Sonar

- Based entirely upon color segmentation



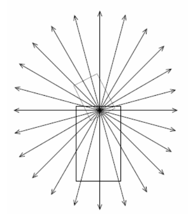


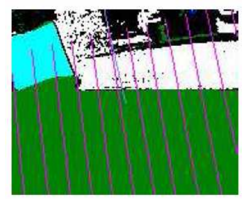
- The main idea:
  - There are only a handful of colors on the field
  - Each color can be associated with one or more objects
    - green -> field
    - orange -> ball
    - white -> robot or line
    - red or blue -> robot
    - cyan or yellow -> goal

http://www-2.cs.cmu.edu/~coral-downloads/legged/papers/cmpack_2002_teamdesc.pdf

Visual Sonar (cont’d)

- Based entirely upon color segmentation



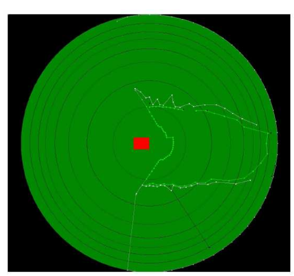


- The main idea:
  - Discretize the image by azimuth angle
  - Search in the image from low elevation angle to high for each azimuth angle
  - When you hit an interesting color (something not green), evaluate it
  - You can infer the distance to an object for a given azimuth angle from the elevation angle and the robot geometry

http://www-2.cs.cmu.edu/~coral-downloads/legged/papers/cmpack_2002_teamdesc.pdf

Visual Sonar (cont’d)

- By panning head you can generate a 180°+ range map of the field

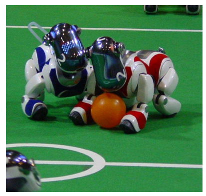
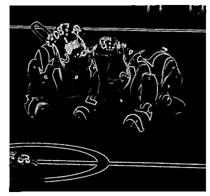


- Subtleties:
  - Identifying tape
  - Identifying other robots???
  - Advantages over IRs
- Video Link

http://www-2.cs.cmu.edu/~coral-downloads/legged/papers/cmpack_2002_teamdesc.pdf

A Similar Approach…

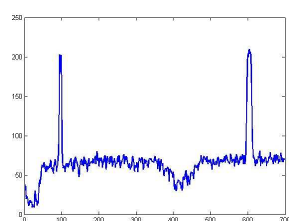
- Based entirely upon edge segmentation

- The main idea:
  - All edges are obstacle
  - All obstacle must be sitting on the ground
  - Search in the image from low elevation angle to high for each bearing angle
  - When you hit an edge, you can infer the distance to an obstacle for a given bearing angle from the elevation angle

Why Does this Work?

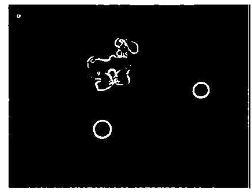
- Recall that edges correspond to large discontinuities in image intensity

- While the carpet has significant texture, this pales in comparison with the white lines and green carpet (or white Aibos and green carpet)

Why Does this Work? (cont'd)

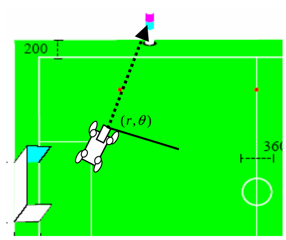
- Let's look at a lab example

- OK, that did not work so great because we still have a lot of spurious edges from the carpet that are NOT obstacles
- Q: How can I get rid of these?
- A: Treat these edges as noise and filter them.
- After applying a 2D gaussian smoothing filter to the image we obtain…

Position Updates

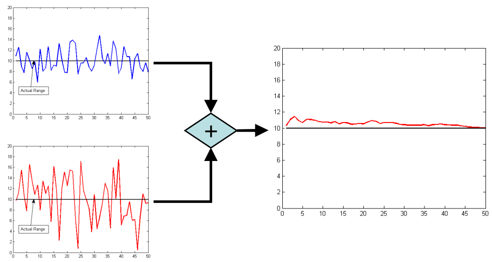
- Position updates are obtained using the field markers and the goal edges
- Both the bearing and the distance are estimated to each field marker.



- To estimate the pose analytically, the robot needs to view 2 landmarks simultaneously
- CMU uses a probabilistic approach that can merge individual measurement updates over time to estimate the pose of the Aibo
- We will discuss this in more detail later in the course

The Main Idea…

- Let's say instead of having 2 sensors/sensor model, we have a sensing and a motion model



- We can combine estimate from our sensors and our motion over time to obtain a very good estimate of our position
- One "slight" hiccup…

The Kidnapped Robot Problem

4.2 Standard Removal Penalty

Most infractions in this league result in the removal of the infringing robot from the field of play for a period of time. This process is called the standard removal penalty. When the head referee indicates a foul has been committed that results in the standard removal penalty, the assistant referee closest to the robot will remove the robot immediately from the field of play. The robot should be removed in such a way as to minimize the movement of the other robots and the ball. If the ball is inadvertently moved when removing the robot, the ball should be replaced to the position it was in when the robot was removed.

The operator of the GameController will send the appropriate signal to the robot indicating the infraction committed. If the wireless is not working and the penalty is timed, the assistant referee handling the robot will reset the robot into the *penalized* state for the duration of the penalty. This is not done if the penalty is not timed, i. e. it is a 0 seconds penalty. After a penalty was signaled to the robot, it is not allowed moving in any fashion, such as being in the *initial* state. The removed robot will be placed outside of the field facing away from the field of play. The best option is to put them on robot stands that are located behind the goals.

The GameController will keep track of the time of the penalty. The operator of the GameController will signal the assistant referees when the penalty is over, so that one of them can put the robot back on the field. The assistant referee will then place the robot on the field on the halfway line as close to the sideline as possible. The robot should be pointed towards the opposite sideline. The robot should be placed on the side of the field furthest from the ball. If there is another robot already in this position, the robot should be replaced at a nearby location along the sideline facing towards the opposite sideline. If there are no practical locations nearby, a location along one of the sidelines should be found that is away from the ball (the robot should be set down facing

- If you are going to use such probabilistic approaches you will need to account for this in your sensing/motion model

Summary

- We reviewed much of the sensing & estimation techniques used by the recent CMU robocup teams
- Complete reliance on the vision system – primarily color segmentation
- Newer approaches also rely heavily on line segmentation - we may not get to this point
- There is a lot of “science” in the process
- There are a lot of heuristics in the process. There work well on the Aibo field, but not in a less constrained environment
- Approaches are similar to what many other teams are using
- Solutions are often not pretty - often the way things are done in the real world