

CSE 398-011  
Exercise 6:  
Lab Date:

Robocup  
Ball Tracking in World Coordinates  
13 Mar 06



A. **Objectives:** The goal of this exercise is to leverage our work in color segmentation, kinematics, and camera calibration for the Aibo to develop a behavior that will track the position of the ball in 3D coordinates with respect to the robot's body frame. Your robot should search for the orange ball, and when it is detected report back its relative position with respect to the robot's base/body frame.

**B. Ball Pose Estimation:**

- a. You are to implement your own `TrackBall` behavior. You can use the `ChaseBall` behavior as a basis for this assignment. However, your ball **MUST** return a position estimate of the ball in 3D coordinates with respect to the robot's body frame.  
<http://www.cs.cmu.edu/~tekkotsu/dox/>
- b. Your behavior will need to wait on a `VisionObjectEvent` to obtain position information for the ball. There are a number of useful functions for obtaining geometric information of the ball (e.g. `getTop()`, `getBottom()`, etc.).
- c. For this exercise, there are two ways to estimate the relative position of the ball with respect to the camera frame.
  - i. Using the elevation angle to the ball.
  - ii. Using the size of the bounding box to infer its distance.

Both of these will rely upon the ground plane constraint (that the ball is actually resting on the carpet).

- d. We will assume a perspective camera model for this lecture (for now) ignoring radial and tangential distortion coefficients. Recall that to project points in the world to points on the image plane, we have

$$x_{cam} = f \frac{x_w}{z_w} \quad y_{cam} = f \frac{y_w}{z_w}$$

And to convert from image plane coordinates to pixel coordinate we have:

$$x_{cam} = -(x_{im} - o_x) s_x$$
$$y_{cam} = -(y_{im} - o_y) s_y$$

NOTE 1: The camera parameters that you obtained through the calibration process are  $f_x=f/s_x$  and  $f_y=f/s_y$ , and  $(o_x, o_y)$  is the image center

NOTE 2: BE CAREFEUL when transforming across coordinate frames. The above formulas work for the convention that the coordinate frame for the *camera* coordinates is +z out the optical axis, +y up, and +x to the left (where the origin is the optical center). The origin for the *image* coordinates is the upper-left corner of the image, and +x is to the right and +y is down. IF YOUR COORDINATE FRAMES ARE NOT CONSISTENT, YOUR CALCULATIONS WILL BE WRONG.

### C. Kinematics & Frame Transformations:

a. Review the kinematics section of the Tekkotsu tutorial:

<http://www.cs.cmu.edu/~tekkotsu/Kinematics.html>

b. We will use the frame transformations from the camera frame to the body frame to estimate the position of the ball with respect to the Aibo's body frame. To do this, you should review the `Kinematics` class reference.

<http://www.cs.cmu.edu/~tekkotsu/dox/classKinematics.html>

c. There is a global instantiation of Kinematics named `kine`, which can be used to perform these calculations regarding the joint positions currently in `state`. You can use this to infer the transformations from the base frame to the camera frame using access functions as below.

```
kine->jointToBase(CameraFrameOffset);
```

d. Once you obtain the transformation matrix across the coordinate frames, you can estimate relative position of the ball. You will probably need to review the class notes on coordinate transformations, homogenous coordinates, etc.

[http://www.cse.lehigh.edu/%7Espletzer/cse398\\_S06/lec003\\_Kinematics.pdf](http://www.cse.lehigh.edu/%7Espletzer/cse398_S06/lec003_Kinematics.pdf)

e. Again, recall that if you would like the correct position, *you need to ensure that your coordinate frames are consistent with what the Aibo is expecting, and that your results will be respect to the base frame defined in the Aibo.*