**PiCar V for ball tracker**  
Abstract-There is a set size that the camera can take in due to its ability to physically manipulate the angle it is viewing. The camera scans for a blob. If it finds one that is smaller than the desired size (signifying that it is far away, and we must move forward) based on the angle the camera had to pan, out vehicle will turn that angle. This makes the vehicle always scanning and since it uses the same angle as the wheel angle it will simply adjust as needed.

main()

we have 3 default angles that are used to determine the camera pan, tilt and front wheel angle. Then we call find\_blob() which will locate the blob for us which returns our blob parameter including a radius. We then check to see if we are larger than the min size to make sure we have detected something. If we don’t have a radius greater than min size we go into a loop to pan and tilt the camera. If the radius is less than max size we have 4 different if statements to determine the location of the blob and alter the needed tilt/pan angles. After this logic loop we set the front wheel angle and based on out pan angle we calculate the need for a turn. Then we simply engage forward or backwards as desired.

find\_blob()

This is the other main function that requires some explanation. We create out radius and set it to . Then the read function brings in the image from the camera and we duplicate the image to have 2 copies to compare to. We convert the image to HSV then use the inRange() function to keep only the pixels we want. They use red but we could alter as needed. They do this with 2 similar hue ranges then combine the images to have one crisp image to use. They use a function that is called HoughCircles() with detects a circle. We would need to find an algorithm to work for a square. The next step is to loop over all detected circles and outline them in hopes of detecting the greatest outer circle. We then return the center point and radius and relative circle location.

Much of the algorithms and logic are reusable for out purposes. The main things we need to alter is making the findblob function work for out purposes then altering the main such that it is also in accordance to our changes.

**Square Image Recognition**

This paper speaks to the methods of analysis and concepts on developing algorithms that can recognize a square in the facet of which we desire. The first mentioned method is that of segmentation. We partition the digital image into disjointed regions then after this we can measure the objects and classify them. Scale space is the next method which treats the scale as a continuous variable rather than parameter. According to the pater the algorithm is as follows:

“1- Plot the density curve with initial bandwidth (automatic bandwidth), ba.

2- Set a limited sequence of bandwidth values with initial bandwidth in the middle of the sequence B.

3- Plot the density curve with each value in the sequence, and count numbers of minima (thresholds), which resulted from each bandwidth, and store the result in vector, yu.

4- From the sequence B, find the bandwidth values corresponding to the values, which have most frequencies in the vector yu, store the results in vector Bu.

5- The optimum bandwidth is obtained as a median of values in Bu.”

After the next method is the median filter method which will smother noise of the image removing things we don’t need while remaining simple and efficient. The last method is the projection method which sums values along image rows.

These several different methods are the basic building blocks of various concepts in digital image recognition. It seems like the default PiCar uses segmentation and for our purposes will most likely be enough. The author suggests using multiple different methods in unison in order to create the best possible result but the basis of the algorithms we have should work for us.