James Hulme - s0901522

SDP - Group 7 - Performance Review 2 - February 15, 2012

For this milestone I attempted to improve the accuracy of the robot detection and to enable the vision system to detect orientation. I was partially successful in this: achieving an increase in robot detection accuracy.

After the last milestone the vision system used hard coded thresholds these were effective but it meant manually changing the values of the threshold depending on lighting conditions. This turned out to be easy for the blue robot, green plate and ball thresholds, but awkward to do for the yellow robot and grey circle thresholds. So I integrated the old code, so now we use the click to threshold for the yellow robot and both grey circles.

The problem with robot detection from milestone one was that there could be noise from the other robots, for example white parts on robots often match the yellow thresholding, this meant that the centroid calculated was incorrect. This is obviously a problem so I researched and evaluated various techniques such as improved thresholding, blob extraction, hough transform and region growing.

The advanced thresholding (method found here `http://en.wikipedia.org/wiki/Thresholding_(image_processing)`) was effective on the yellow, but very slow: reducing the framerate from 40fps to 15fps doing just one iteration. It was decided to use this technique as a backup in case the vision system lost the robot, but I was unable to come up with a method for recalculating the difference between two colour channels when rethresholding the image which is necessary for the blue robot.

I then implemented a technique which only checked for matching pixels within a certain range of the previous centroid, this works because the user defines the initial centroid. This was fast and effective but if the robot moved too fast or if the T was covered then the robot was lost. This posed a huge problem for when the robots have to be reset mid game as we cannot guarentee the opponent team will not cover their T whilst resetting the robot. This is when the advanced thresholding would have been used.

I then used blob extraction (method found here `http://en.wikipedia.org/wiki/Blob_extraction`), initially with 4 connectivity, then 8 connectivity and finally 'two layer four connectivity' which is four connectivity on four connectivity on the original point, this means that a wide area around a pixel is checked, eliminating the small areas of noise, before including it in the centroid calculation. In order to keep it fast I had to leave off the second pass to cluster the points, so some small bits of noise do still appear but the size of them has been reduced so they affect the centroid less. This is the technique that is currently used in the vision system.

Figure 1: Diagram of the pixels checked in the two layer four connected blob extraction method.

To test the accuracy of the different robot detection techniques I used a by eye technique to tell how improved it was, I used a similar situation for each technique and was able to tell how close the calculated centroid was to the centre of the robot.

For finding the orientation I designed a method to find the points of the T that lie on the circumference of a circle drawn around the robots centroid. This was showing promising results, but needs more work doing to it which will be done after the milestone two.

I believe that I have been mostly successful in my goals for this milestone. For the next milestone I will be focusing mainly on detecting the orientation. I will also help finetune what we have now and also probably do some programming with the strategy team.