SYSTEM DESIGN PROJECT 2012

Milestone 2 Individual Report Group 7 – s0951580

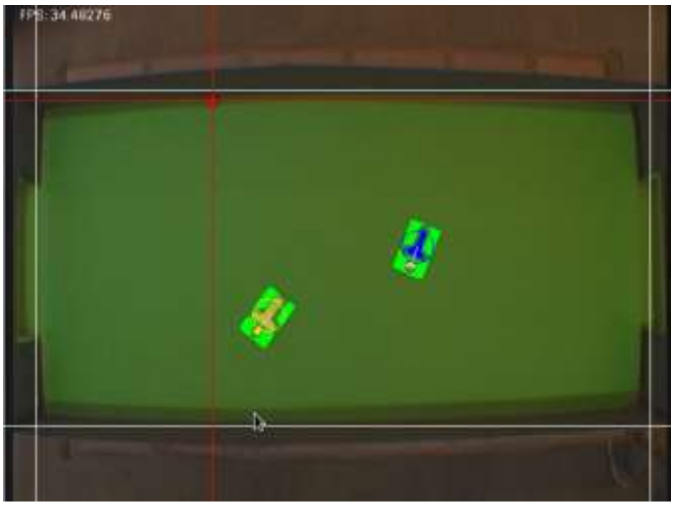
INTRODUCTION

Throughout the past two weeks I have been heavily involved in implementing and experimenting with various methods to find the orientation of the robot. Two of the methods involved the grey circles and two of the methods I worked on involved the green plate. Frustratingly, none of these was sufficiently accurate to satisfy the robustness needed for this project. The goal was to have the orientation; however, this was not achieved.

GREY CIRCLES

Using RGB thresholding for the grey colour has proved very challenging. The first attempt at finding the orientation using the grey circles involved thresholding the imaged for grey colour and getting the centroid of the bulk of grey pixels in a sufficient radius around every robot's centre. However, due to the shadow cast by the robot itself, this wasn't working at all.

In order to solve this problem I decided to implement James's idea to have a dynamically recomputed centre of the grey circle. The initial centroid is taken from a click at the start of the system and grey pixels in a certain range from the robot's centre are taken. Furthermore, only pixels in a certain range around the grey circle's centroid were taken. Background removal was also used in a radius around the robot to remove any unnecessary grey pixels. All of this was done with hope to isolate the grey circles and to detect only them. It worked with slow movement of plates:



However, once the robots were moved realistically, the grey circles were lost. Improvements could be made but, given that the majority of last year's groups did not used grey circles, I decided to try a different approach.

Green Plates

The thresholding for the plates is much easier due to the high value of the green channel of the object. My first approach was to find the centroid of green pixels within a certain distance from the robot's centre. However, due to barrel distortion at the corners of the pitch a lightly bigger number of green pixels were picked up on the side closer to the camera. So this method was unreliable at the corners.

The second method was to find the smallest bounding rectangle around the green plate. This method gives four "end" points of the plate. The two points with smallest distance to the robot are picked because this should be the front two points. Their average point gives the point directly in front of the T. This method proved to be accurate but only most of the times. It was very jumpy and for the different positions of the green plate it was mistaking which were the closest points. For example the correct angle was picked always but several other values were picked as well. This might be improved using knowledge about the robot's T and checking for the bounding points closest to its points. This would require an algorithm for finding significant points of the T.

Summary

Although none of these methods produced the required highly accurate result for this task, much in terms of the approach to the problem was learned. For example I will now concentrate on using the T's points to extract the orientation. This might include calculating orientation moment, or using a suitable library to extract the necessary angle of orientation. The goals for the next milestone would be to have a robust and fairly accurate vision system, which would produce the necessary accurate data.