SYSTEM DESIGN PROJECT 2012

Milestone 3 Individual Report Group 7 – s0951580

INTRODUCTION

Throughout the past three weeks, I have manly been involved into the development of the vision system and I also was involved in the angle system and the implementation of some of the control functions for the milestone. My main contributions for this milestone was getting the vision system to work on both pitches, implementing an algorithm for barrel distortion, one for looking for points inside the green plates, and the unification of the angle system.

Vision

Following from my last report, we needed a more reliable method to find the angle of orientation. I decided and proposed to try and make JavaCV working. It turned out that the C code running OpenCV wasn't able to instantiate the camera and get the feed. I then tried to have the V4L4J library and convert the image to suitable type of image for manipulation by JavaCV. However, this wasn't successful as a weird error was thrown at the conversion. Before I could further explore the problem, Dale and his ingenious solution for finding the angle made it not necessary.
I then went on to isolate the threshold pixels of the robots' Ts from any possible noise in the image. Using some of the functions of Dale, I checked for pixels only inside every robot's green plate. This solved the problem that sometimes, mainly for the yellow robot, the grey parts outside the plate were picked as most distant from the centroid of that robot, thus as the direction of the robot.

I then used a method implemented by team 5 form last year. It takes into account the relevant position of the pixel to the height and width of the screen, as well as to the centre. The new position of the pixel is then calculated using distortion coefficients that compensate for the distortion effect of the camera. The outer a pixel is, the outer it is stretched to fix the distortion. The method was reducing the frame rate from 30 fps to 10 fps, so it was James who went on to apply it only to the pitch area, which produced 20 fps. I had to then refine the vision system for the second pitch in terms of constants of the pitch boundaries and thresholding values, as the lighting conditions there are different.

Control

Before the first friendly I was trying to produce a hack as our arcing wasn't working. I failed, but this led to the discovery that control and vision weren't using the same angular system. One the group meeting Tom, Darie, Victor and I were appointed to solve the problem. I discovered that one of Tom's functions to find the actual angle, so that the robot can turn, was using some normalizations and transformations that were messing with the angle. I proposed to use just the same conversion used by the vision system and that solved our problem with the angles, I hope for good.

I also helped towards this current milestone, as collectively with Tom implemented a new version of the code for the milestone. It substituted Darie's change of slope method of the path, with change in the vector between points. This was much more robust and by physically testing its behaviour, we concluded that the method was much more reliable.

Conclusion & Goals

The work on the vision system is nearly finished. James has already started testing and I will probably introduce sliders to make it more convenient for everyone to use it. Also produce some guidelines how to use it. I would probably get involved into other areas, such as Control or Planning, to help towards our performance.