# System Design Project 2012

Milestone 3 Individual Report
Gediminas Liktaras (s0905195)

March 6, 2012

## Goals

For the third milestone I kept working on the vision component of the robot system. By milestone two the performance of the vision was tolerable - robot position could be reliably found but the direction was still finicky. My goals were to improve the system's direction detection and implement a host of other features, such as barrel correction.

## Direction Detection

I have implemented and tested two additional direction detection procedures.

### Largest Slice

After finding the mass center of the blob that represents the robot's T shape, this procedure scans the distance from the center to the edge of the blob in all directions. This results in a histogram of sorts.

To find the direction of the robot, the procedure then examines the histogram values in continuous blocks of 30 and looks for the one with the largest sum. This loosely corresponds to finding the largest 30 degree "slice" in the shape blob. Since the longest spike of the "T" is pointing in the same direction as the robot, the direction of the largest slice is what we are looking for.

### Shape Matching

A bitmap with a template of the T shape is overlaid over the blob's mass center. The procedure then measures how many pixels match in the blob and in the shape. This is done for every orientation of the shape and the direction with the most matching pixels is said to match the actual robot direction.

## Test Bench

Together with Aaron, we have finished implementing the vision test bench. Now it can provide objective measures of the system's performance, which include positions and directions of all three objects on the field.

## Barell Correction

I have employed OpenCV to undistort incoming frames. OpenCV helpfully contains all the functionality needed to extract the necessary data from the scene and undistort the frame and implementation of this feature was relatively simple.

## Areas That Need Improvement

### Variable Thresholding

The biggest problem with the vision system at the moment is varying lighting conditions across the field. Although static thresholds are adequate most of the time, in a few instances the system produces inaccurate results – when the shape is "bleeding" (more is thresholded than needed) for instance.

This could be solved with threshold maps – data structures that provide unique thresholding values for every pixel of the frame. I have even carried out refactorings to allow for the implementation of this feature, but it is not complete.

### Easier Calibration

Right now all threshold bounds have to be entered and adjusted manually. Parts of this process can be automated. Since this feature shares some logic with and variable thresholding and manual creation of threshold maps would be very cumbersome, it is only natural provide the two together.

## Future Work

It is the team's consensus that the vision system is good enough for our purposes. Therefore, I will switch to the AI and work on improving our robot's behaviour on the pitch. There are quite a few things I would like to add to the vision system, but I cannot afford this luxury due to time constraints.

## Expected Score

6 points.

## Vision System Performance