System Design Project

Individual Report **3**

Aleksandar Krastev – Group **12**

Problems encountered in Vision system

For this milestone, the vision sub-team wanted to finalise the vision system. This was crucial, because the strategy could not be tested and proved reliable.

At one point during testing, nothing was working as supposed to. The robot was moving randomly in different directions. I realised that the problem lies either in the vision or strategy. Therefore, I proposed to look at the angles our system was receiving and see if their instability is affecting the behaviour of the robot. After looking at the Vision GUI, Behzad Tabibian constructed, I found out that the inaccuracy of the angles made the strategy navigate the robot in a zigzag movement. This helped us realise where we need to work more.

It was clear that the vision had to be reworked so that it outputs angles, which are not jumping in a small range. Moreover, the strategy had to ignore small changes in orientation. This meant that our team had to sacrifice some precision. Until me and Behzad started working on a method, which allowed us not to limit the accuracy when the robot turns.

Different methods considered

Behzad implemented a background subtraction method, which subtracts an image of the pitch from the current frame. Thus, removing some noise, which hinders our algorithm from finding clear objects. I tested this method and I concluded it is of good use to our team.

Next, I started testing the methods for calculating orientation we had so far. For the previous milestone we used the centre of the plate and the T shape to get the angle. I tried this, but it was not robust enough, because even with small light changes, it started outputting noisy values.

Behzad researched the usage of image moments[1] and impelemented the mathematics needed to calculate them. We started by using first order of cental moments to calculate the orientation. However, like the previous method it was not good enough. This was because, this order of moments only finds the centre of the T shape object. Combined with its centre of mass, it finds an angle using the function atan2[2].

Winning solution

A long time ago, we had implemented calculation with second order of central moments, which takes into account the orientation. However, due to insufficient testing this method was dropped quickly. Me and Behzad decided to give it one more try. This time with better thresholds for the T shapes and background subtraction, we were pleasantly surprised at the accuracy and stability of it [Fig 1]. After playing with it and averaging out fluctuating values for the angles, we were able to deliver a stable output for the strategy to work without any limitations.

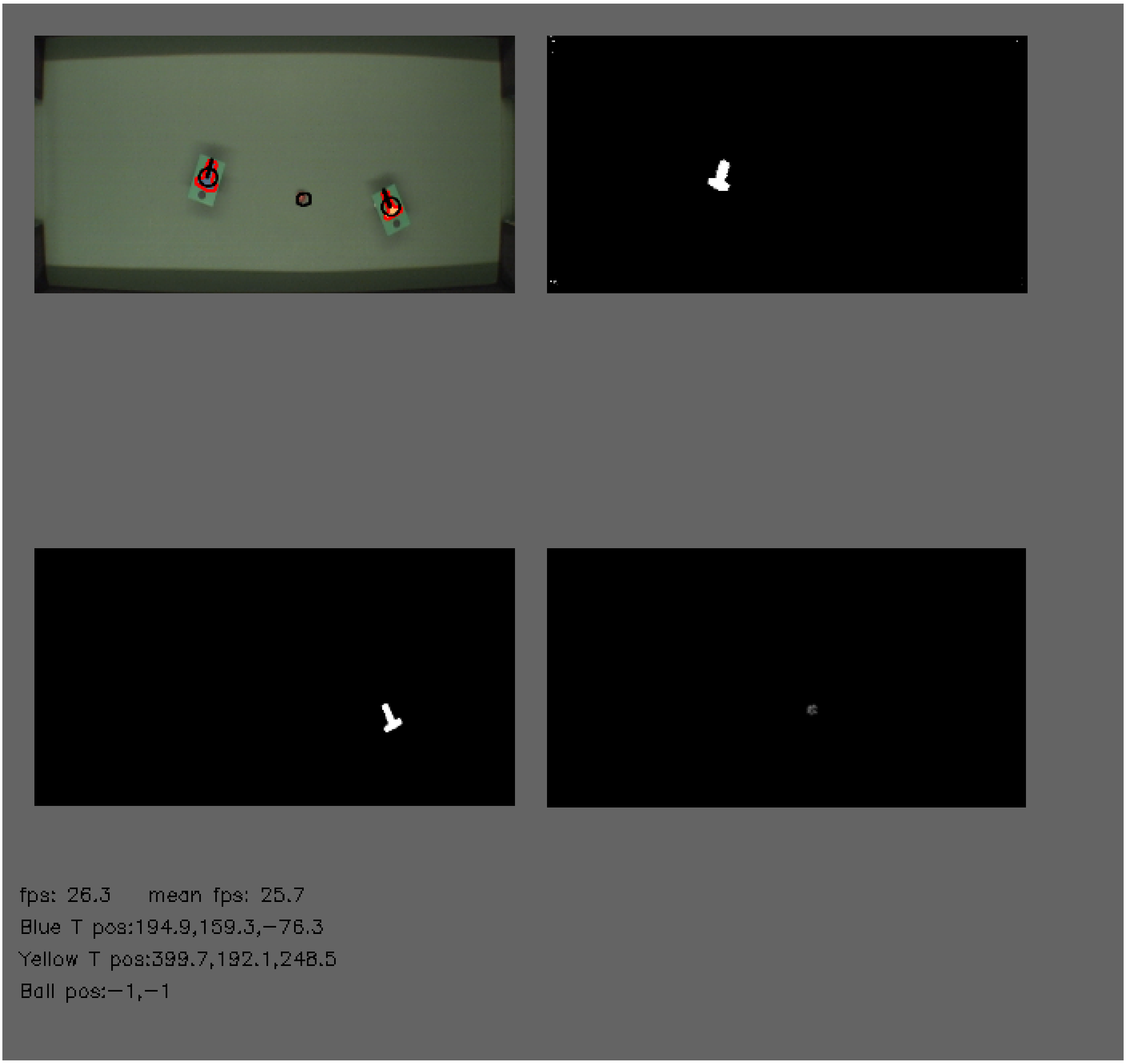
Future work

The second order of moments works in 90% of the cases [Fig2.]. I only swapps the angle by PI, when the robot is facing angles around PI/2 or -PI/2. We have not yet came with a solution, but hope to make our orientation method the most robust of all other team's approaches.

[1] http://en.wikipedia.org/wiki/Image_moment

[2] http://www.cplusplus.com/reference/clibrary/cmath/atan2/

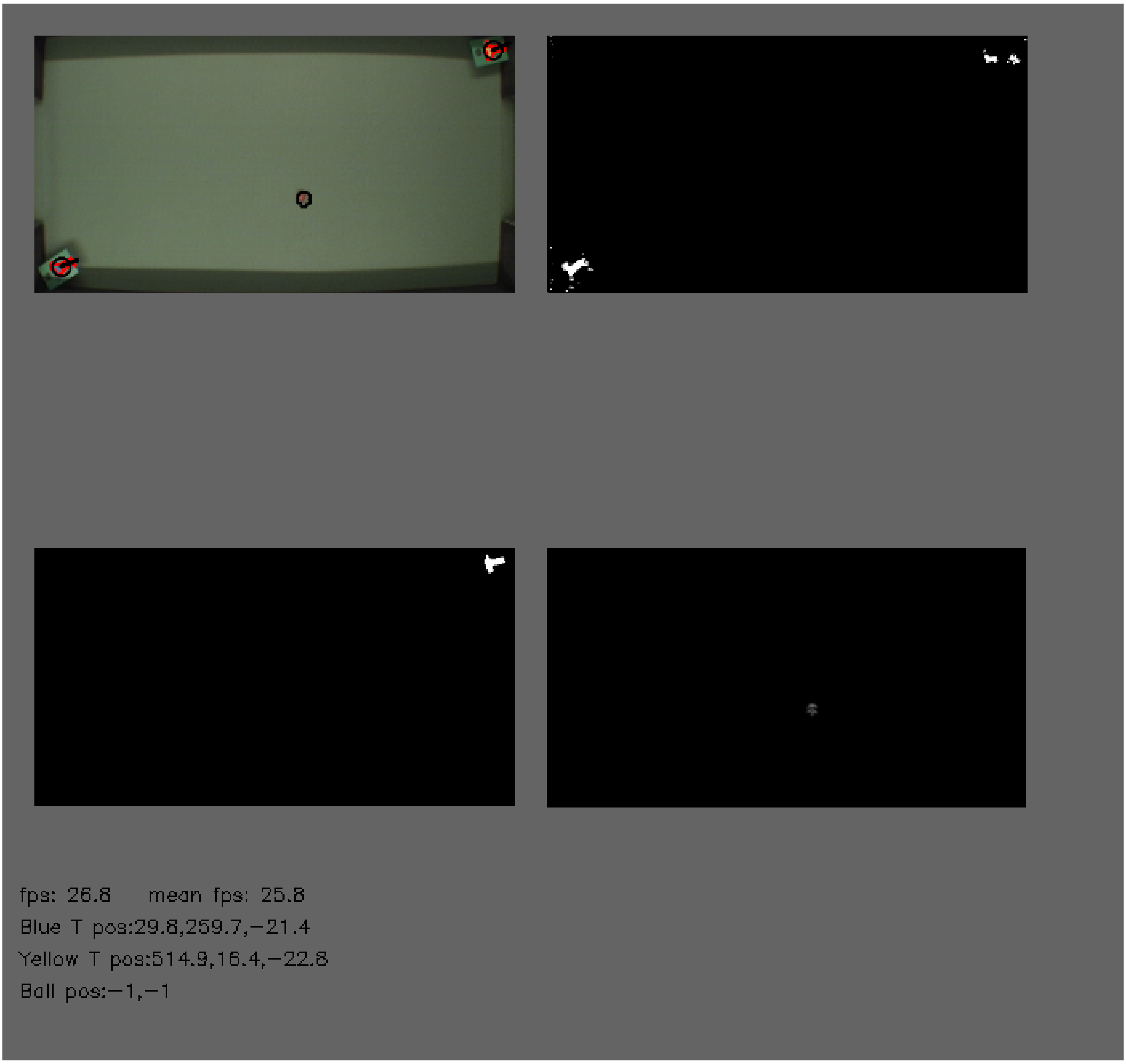
Appendix



```
fps: 26.3    mean fps: 25.7
Blue T pos:194.9,159.3,-76.3
Yellow T pos:399.7,192.1,248.5
Ball pos:-1,-1
```

Fig1. GUI of the Vision system. Orientations are drawn on the upper-left image

1. http://en.wikipedia.org/wiki/Image_moment
2. http://www.cplusplus.com/reference/clibrary/cmath/atan2/



```
fps: 26.8     mean fps: 25.8
Blue T pos:29.8,259.7,-21.4
Yellow T pos:514.9,16.4,-22.8
Ball pos:-1,-1
```

Fig2. Orientation works even in extreme positions under shadows.

1. http://en.wikipedia.org/wiki/Image_moment
2. http://www.cplusplus.com/reference/clibrary/cmath/atan2/