Computer Vision - Project 3
Automatic Darts Scoring System

Miruna-Andreea Zăvelcă

1 Introduction

The goal of this project is developing a program that can automatically detect the darts on a dartboard - number and positions. The project has 3 tasks:

1. Detect darts on a target board
2. Detect darts on a classic board
3. Detect darts on a classic board from a video

Details can be found here: `http://tinyurl.com/CV-2022-Project3`

2 Templates

Since the pictures are taken from the same angle, I used a template for each task in order to find the darts on the board and to infer the position on the board based on the pixels in the image.

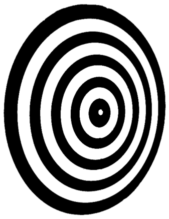

Figure 1: Template for task 1

From this template I inferred the list of polygons (ellipses). Counting from the center outwards, where the center has index 0, *point(x, y)* has score *i* if it is inside *polygon(10 - i)*, but not inside *polygon(10 - i - 1)*.

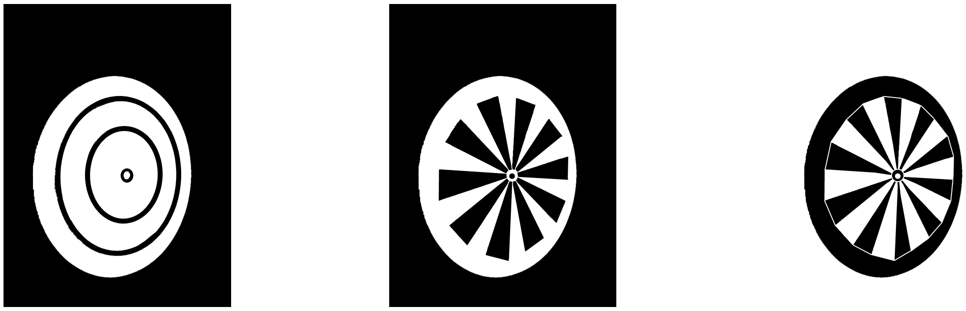

Figure 2: Templates for task 2

Similar to the previous task, I used the first template to identify the simple/double/triple/bull flag, and the regions from templates 2 and 3 to identify the region.

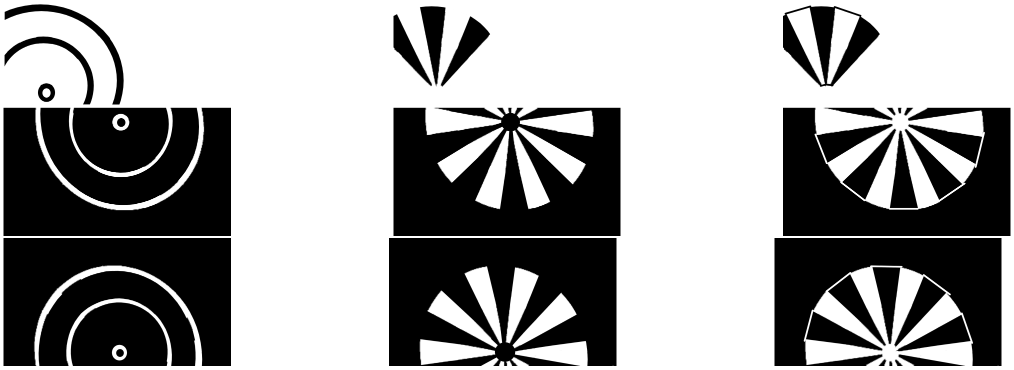

Figure 3: Templates for task 3

For task 3 there are 3 possible angles from which the video was taken. For each of these cases I made a set of templates, following the format presented at task 2.

3 Dart counter

The first 2 tasks require counting the darts on the board. I first changed all images to HSV format, so image brightness will not be an issue. Since all darts have the same red and blue flag, after the transformation the flag becomes preponderantly green. For the first task, this is more or less the only green spot found in the new image, while for the second one there are also some spots on the board. In order to remove them, I composed the absolute difference between the board and the given template before turning it into HSV format. After this, I applied a green mask on the image, so I could keep only the relevant information.

The mask formed this way is a black image with white spots where there is relevant information (dart flags). I used a sliding window (size 100, step 50) to filter and keep only the squares with the percentage of black smaller than a given threshold (40 for task 1, 12 for task 2), and then I rejoined the intersecting squares in order to find the minimum sized rectangles with relevant information. Each one of these rectangles surrounds a dart flag, thus their number represents the number of darts in the picture.

4 Position detection

After analysing the distance in pixels between each flag and its tip, I have computed the median of those distances and decided this is where a dart's tip is most probable to be. With this assumption, I got a 0.6 accuracy on task 1 and 0.38 accuracy on task 2.

For task 3, the frames overlap perfectly. I used the first frame of each video as a mask over the last frame and the resulting image had only the last thrown dart and some noise. I used a color mask for removing the noise and a sliding window to get the rectangle surrounding each dart. The bottom-left point of that rectangle represents the tip of the dart. This approach had a 0.7 accuracy.

The code can be run from *main.py* or from the files for each separate task (*task1.py*, *task2.py*, *task3.py*) and the results will be found in the folder *evaluation*, each task in its own subfolder.