MEE / MEEE

Visão Computacional / Computer Vision

Assignment 1 – Colored ball detection and measurement

1. Instructions

a) **Integrity and collaboration:** Students are encouraged to work in groups, so include the names of your collaborators in your write-up. Code should **NOT** be shared or copied among groups. **Plagiarism is strongly prohibited and may lead to the failure of this course.**

b) **Start early!** Especially those not familiar with Python and OpenCV.

c) **Submission:** Your submission for this assignment should be a zip file, **<CVis_2223_Assign1_JohnDoe_JaneDoe.zip>**, composed of your write-up, your Python implementations (including any helper functions), and results. Please make sure to remove any other temporary files you generated. Substitute both in the folder name and in the filename "**JohnDoe**" and "**JaneDoe**" by each of the group element's names and surnames.

Your final upload should have the files arranged in this layout:

<CVis_2223_Assign1_JohnDoe_JaneDoe.zip>

- < JohnDoe_JaneDoe_ Assignment1>.pdf => this is your report.
- **data** => location of the images that you are using.
- **papers** => folder to add papers with methods applied in the assignment.
- **code_python** => location of the scripts that you will develop. You should **explicitly mention how to run** each file in a **readme.txt** file or similar.

d) **File paths:** Please make sure that any file paths that you use are relative and not absolute. Avoid using complete paths like in the following example: cv2.imread(**'C:/name/Documents/assign1/data/xyz.jpg'**). Use relative paths like cv2.imread(**'../data/xyz.jpg'**) instead.

2. Implementation

This assignment uses as a reference image the **Boccia_balls.jpg**, which displays an example of the colored balls used in the game of Boccia. The game aims to throw these colored (red or blue) leather balls as close as possible to a white target ball, also named jack (not shown in the picture). It can be played individually or in teams, for each round the team with the closest distanced ball regarding the jack wins a point.

For the Computer Vision assignment, the students are challenged with various tasks, such as, segmenting the region defined by the two balls in the image, extracting their measurements, and extracting their distance from the camera. For each of the tasks, students will have to choose, justify and document the best processes to obtain the expected result.

Every script and function that you write or use in this section should be included in the **code_python** folder and **explicitly mention how to run it**. Please include the resulting images of the entire process in your write-up. Also, if the techniques/methods used don't belong to OpenCV's native set functions, students must document the sources they used (add the paper pdf file in the **papers** folder), indicating the relevance for their use in this context.

2.1 Segmentation of the Boccia balls applying grayscale conversion

This task aims to extract the region of interest (ROI) individually for each of the Boccia balls present in the aforementioned image. In this task, it is proposed to use a greyscale conversion of the colored image, and the usage of filtering processes to obtain the contours of the ROIs. From these external contours, you may obtain a binary mask image, for both regions. You can compare your result against the ground-truth image[1] (**Boccia_balls_GT.jpg** in the data folder), applying the Dice Coefficient. You can also determine the bounding boxes that enclose these ROIs, and plot them over the original image, with the measurements of the object (width and height).

2.2 Segmentation of the Boccia balls applying color space conversion(s)

Similarly, to the previous task, the objective is to extract the ROIs for the Boccia balls, obtaining such extraction from color features. It is proposed the use of different color spaces that can aid in this task and get the binary mask image for both regions and compare it against the ground-truth image as well.

[1] The ground truth image is stored as a 3-channel image. You will only need one of the channels, as binary image, when reading the image in your script.

2.3 Distance measurement estimation regarding the camera

The ball in the red color is placed 40 cm from the camera and has a diameter of 8.2 cm. Calculate an estimation of the focal distance for this camera. From that result, infer the distance for the red ball regarding the camera.

2.4 The real challenge: Run the algorithm in a video

Gather all the previous methods and run them on a pre-recorded video (**Boccia_video.mp4**, also available in the data folder), to track all the changes frame by frame.

3. Write-up

For each of the previously mentioned tasks, make sure they run correctly and generate intermediate output images. Include the set of intermediate outputs in your write-up and **be critical** of the results you obtained. Did the code work well on all the images with a single set of parameters? How did the optimal set of parameters vary with images? Which step of the algorithm causes the most problems? Did you find any changes you could make to the algorithm to improve performance? In your write-up, you should describe how well the code worked on different images, what effect the parameters have, and any improvements that can make it work better.

Table 1: Classification weights

Activity 2.1	Activity 2.2	Activity 2.3	Activity 2.4
25%	25%	25%	25%