Università
della
Svizzera
italiana

**Faculty
of Informatics**

Computer Vision & Pattern Recognition

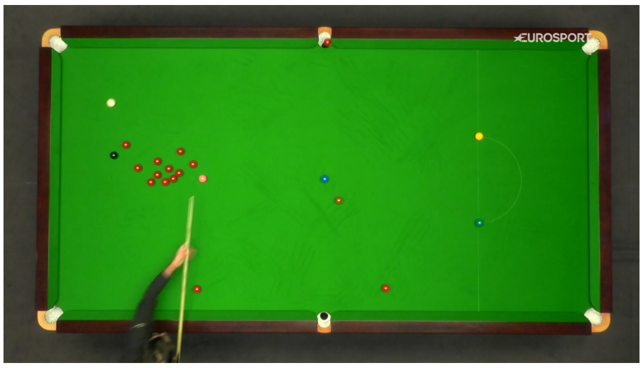
Spring 2022

Assignment 7

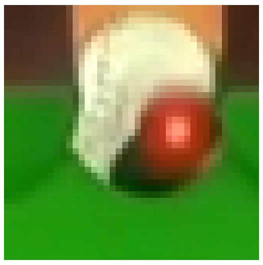
May 19, 2022

Exercise 1 [15 points]

This exercise follows on from the activities of Exercise 1 in Assignment 4 where you wrote a program to locate both the boundary of a snooker table and the red snooker balls. You are to adapt and/or extend this program to automatically detect when a red ball has entered a pocket.
For this task, you shall use the video file: `scottish-open-snooker-2020-osullivan.mp4` (link). You may assume that the camera is in a fixed top-down position and provides a sufficiently 'orthogonal' projection. It is therefore only necessary to automatically locate the table's boundary once, for a single frame.
To locate the pockets automatically, you can combine the detected table boundary and your knowledge of a regulation snooker table's dimensions.
To detect when a red ball is potted, within reason, you may implement any algorithm. Some useful observations are that the balls of interest are always red and that the snooker player and referee frequently obstruct the pockets. Additionally, the ball will be in motion near a pocket when potted. When in motion, the ball can move at high velocities but appears to halt when it falls into the pocket.



(a) Input image



(b) Zoom in

Figure 1: An example of a red ball being potted in a side pocket (a) visible in the upper middle of the image and (b) magnified.

Your program should write an image of the frame to a file, highlighting the pocket where the red pot is detected. The name of the file should indicate the frame number, *e.g.*, `frame_0123.jpg` (N.B. if you are skipping frames, be careful to label these correctly). Only one frame of each pot needs to be written (there are 15 red balls that are potted, so 15 images). The exact frame number is not too important, as long as it is close. Also, your implementation does not need to run in real-time.

With respect to Exercise 2, which follows, consider designing your computer vision pipeline such that it is robust to scene rotations.

Exercise 2 [10 points]

The following exercise shall evaluate the robustness of your method from Exercise 1 to rotations. If your implementation is already invariant under rotation, then you might not have to adapt your code at all.
In addition to the overhead video (`scottish-open-snooker-2020-osullivan.mp4`), three additional videos can now be accessed where the scene has been artificially rotated (link). The file names indicate the number of degrees the video has been rotated by, *e.g.*, `rotated-010.mp4` is rotated by 10°.
For this exercise, you may assume that the position of the camera and the table remain fixed throughout the entire video. Similarly to Exercise 1, it is only necessary to automatically locate the table's boundary once per video.
For each video, you are to conduct a similar experiment to that of Exercise 1—identifying frames where a red ball has been potted. In your submission, as well as the frame number, please note the rotation in the file/folder name, *e.g.*,
`rotated-045/frame_0123.jpg`.

Please hand in your code and the resulting images from processing each video.

Solutions must be returned on June 2, 2022 via iCorsi