University of Bucharest
Faculty of Mathematics and Computer Science
Department of Computer Science

Iuliana Georgescu
Bogdan Alexe
MSC in Computer Science
May 2020

Computer Vision - Project 2

Video analysis of a snooker footage

Objective

The goal of this project is to develop an automatic system for video analysis of snooker footages. The system should be able to detect the snooker table and the balls on the table, track the balls, infer when a ball is potted into a pocket.

Description of the snooker game[1]

Snooker is a sport played on a rectangular table covered with a green cloth (or ”baize”), with pockets at each of the four corners and in the middle of each long side. Using a cue stick and 22 balls (one white, fifteen red, one yellow, one green, one brown, one blue, one pink, one black), players must strike the white ball (or ”cue ball”) to pot the remaining balls in the correct sequence, accumulating points for each pot.

The objective of the game is to score more points than one’s opponent by potting object balls in the correct order. At the start of a frame, the balls are positioned as shown in Figure 1 and the players then take turns hitting shots by striking the cue ball with the tip of the cue, their aim being to pot one of the red balls into a pocket and thereby score a point. If the striker pots a red ball, he or she must then pot one of the six ”colours” balls. Here, ”colour” ball means a ball with a different color than white and red, so this includes

[1]Information taken from Wikipedia: https://en.wikipedia.org/wiki/Snooker

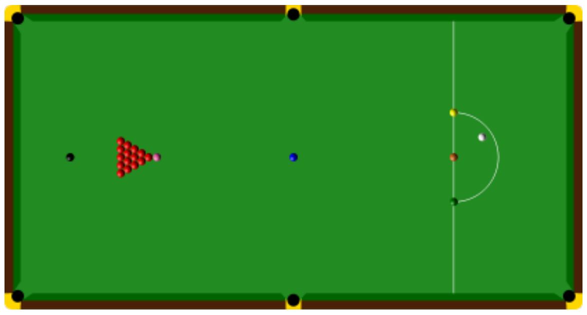

Figure 1: *Snooker table with balls placed in their starting positions. At the start of the game, the cue ball (white) may be placed anywhere in the semicircle, known as the ”D”.*

yellow, green, brown, blue, pink and black. If the player successfully pots a colour ball, the value of that ball is added to the player's score, and the ball is returned to its starting position on the table. After that, the player must pot another red ball, then another colour, in sequence. This process continues until the striker fails to pot the desired ball, at which point the opponent comes to the table to play the next shot. The game continues in this manner until all the reds are potted and only the six colours are left on the table. At this point the colours must be potted in the order from least to most valuable ball as follows: yellow ball - 2 points, green ball - 3 points, brown ball - 4 points, blue ball - 5 points, pink ball - 6 points, black ball - 7 points. The value of a red ball is 1 point.

The oficial rules of snooker are here `https://www.wpbsa.com/wp-content/uploads/2016/03/official-rules-of-the-game.pdf`. A nice video with explanations of the rules is here: `https://www.youtube.com/watch?v=CjSWUTkupQo`.

Data description

The release data directory (available here `https://tinyurl.com/CV-2020-Project2/`) contains the directory training_data. This directory contains data organized in 4 sub-directories, corresponding to the Tasks that you need to solve (see below). For Task 1 there are 50 images of a snooker table seen from above. For Task 2, Task 3 and Task 4 there are several videos, each of them taking around 10 seconds and containing a small footage of a player making a shot (hitting the cue ball).

Tasks

Your job is to write a program in Python that solves three tasks: (1) counting the number of balls on the table from an image and specifying their color; (2) analyzing a video and deciding whether a ball is potted into a pocket and if so recognizing the color of the potted ball and the pocket (one of the six pockets) where the ball was potted; (3) tracking the cue ball and another specified ball in a video (the initial bounding boxes of the two balls to be tracked are provided for the first frame).

This project worths 3.5 points but you can gain a bonus up to one point for a total of 4.5 points.

- **Task 1** - you receive a test set of 57 images of a snooker table seen from above at a random time of a game (for example the black ball might be potted). The task is to count the number of balls on the table and specify their color for each of the 57 images in the test set. By solving correctly this task for an image you will earn 0.02 points as follows: you will receive 0.01 points for counting correctly the number of balls on the table and 0.01 points for correctly specifying the number of balls of each color. For a test image your output should be similar with the one provided in the directory `training_data/Task1/ground-truth/`. **(1.14 points)**
- **Task 2** - you receive a test set of 25 videos containing a player making a shot. In each of the 25 videos the snooker table is seen from above. The task is to decide whether

a ball was potted into a pocket and if so recognize the color of the potted ball and the pocket (one of the six pockets) where the ball was potted. By solving correctly this task for a video you will earn 0.04 points as follows: if the correct answer is NO (no ball was poted) you will receive 0.04 points if your output is NO, else you will receive 0.02 points if your output is YES, 0.01 points for specifying the correct pocket (1 - top left corner, 2 - top right corner, 3 - bottom left corner, 4 - bottom right corner, 5 - middle left corner, 6 - middle right corner) and 0.01 points for specifying the correct color of the potted ball. In each of the 25 videos there will be a maximum of one ball being potted. For a test video your output should be similar with the one provided in the directory `training_data/Task2/ground-truth/`. **(1 point)**

- **Task 3** - you receive a test set of 25 videos containing a player making a shot. In each of the 25 videos the snooker table is seen from above. The task is to track the cue ball (the white ball) and another specified ball. The initial bounding boxes of the two balls to be tracked are provided for the first frame (they follow the format [xmin ymin xmax ymax], where (xmin,ymin) is the top left corner and (xmax,ymax) is the bottom right corner of the initial bounding-box). By correctly solving the tracking problem for a video you will earn 0.04 points (0.02 points for each correctly tracked ball). In each video we will consider that your algorithm *correctly tracks a ball* if in more (greater or equal) than 80% of the frames your algorithm *correctly localizes the ball to be tracked*. We consider that your algorithm correctly localizes the ball to be tracked in a specific frame if the value of the IOU (intersection over union) beetwen the window provided by your algorithm and the ground-truth window is more than 20%. For a test video your output should be similar with the one provided in the folder `training_data/Task3/ground-truth/`. The first frame for which we provide the bounding boxes initialization has index 0, the last frame of a video with N frames has index $N-1$. **(1 point)**

- (bonus) **Task 4** - - you receive a test set of 25 videos containing a player making a shot. Different than the previous tasks, in this videos the snooker table can be filmed from different viewpoints (not only from above). The task is to track the cue ball. The task here is much harder as you have diferent scales of the cue ball, changes in camera viewpoint, the initial bounding box of the white ball is not provided. The initial bounding box of the cue ball to be tracked is provided for the first frame (it follows the format [xmin ymin xmax ymax], where (xmin,ymin) is the top left corner and (xmax,ymax) is the bottom right corner of the initial bounding-box). By correctly solving the tracking problem for a video you will earn 0.04 points. The rules described at Task 3 apply here, meaning that your algorithm should correctly localizes the cue ball in more than 80% of the frames, at each frame correctly localization means that the window provided by your algorithm should have IOU more than 20% with respect to the ground-truth window. For a test video your output should be similar with the one provided in the folder `training_data/Task4/ground-truth/`. The first frame for which we provide the bounding boxes initialization has index 0, the last frame of a video with N frames has index $N-1$. **(1 point)**

- ex officio **0.36 points**;

Deadlines:

Submit a zip archive containing your code and a pdf file describing your approach until Saturday, 13th of June, using the following link `https://tinyurl.com/CV-2020-PROJECT2-SUBMISSIONS`. Your code should include a README file (see the example in the materials for this project) containing the following information: (i) the libraries required to run the project including the full version of each library; (ii) indications of how to run each task and where to look for the output file. On Sunday, 14th of June, we will make available the test set. You will have to run your system on the test set provided by us and upload your results in the same day as a zip archive using the following link `https://tinyurl.com/CV-2020-PROJECT2-RESULTS`. Format your output as the one contained in `training_data/format_output/`.