RuiBot: An Autonomous Snooker Referee

Daniel Carneiro, Felipe Bartelt
Universidade Federal de Minas Gerais
Belo Horizonte, MG, 31.270-901
{dennys, fbartelt}@ufmg.br

Abstract

This paper proposes a method for processing snooker videos to extract the balls positions, their collisions, and colors, with the intention of creating a fully autonomous snooker referee. To do this, first the corners of the table are detected and used to compute a homography, which is then used to warp the table to a fixed size, this allows for consistency when working with diferent videos. Then the most prevalent colors in the remaining image are filtered out to remove the baize from the image. The remaining image is then used to detect the ball countours and their collisions. To classify the color of each contour, we train simple a neural network to classify RGB to one of eight colors, then we use k-means clustering to get the most prevalent color in the contour, and feed it to the trained model to get the color of the ball. Implementation is available at `https://github.com/funnisquares/ruibotchapeu`.

The paper presentation is available at `https://www.youtube.com/watch?v=xFSQhHuhxyo`.

1. Introduction

The world of sports is moving in the direction of techonology, specially when relating to arbitration. Examples of this can be seen in tennis, cricket, football and many others [2]. Solutions vary from simple video replays to computer vision to using microchips in balls for example. However most implementations of such technologies still rely on a human referee to manually analyse the data provided, and make a final arbitration.

In order for a machine referee to be fully autonomous, it is first and foremost necessary for it to require little to no human interaction, and that its arbitrations are consistent and correct. In this paper we will explore some of the challenges of implementing such a system for snooker arbitration, focusing mainly on obtaining the information necessary to decide the outcome of a play.

The content of this paper is organized as follows: Section 2 provides an overview of the related work. Section 3 describes the methods used to extract information from the footage, and shows our results alongside the descriptions, this was done because our experimentations and methods were devised alongside one another. We conclude and discuss future works in Section 4.

2. Related work

2.1. Event Detection in Sports

Event detection in sports [6, 8, 9] consist of the combination of techniques to process sports footage to extract the information needed to detect events. An event can be anything related to the game, such as a scoring play or a foul.

Usually detecting these types of events require knowledge of the game rules (Although there are exceptions such as in the work of Tjondronegoro and Chen[8]), and information about the positions of objects of interest. In the context of snooker, the objects of interest are the individual colored balls and the table. Another important factor in event detection is the ability to determine when an event starts and when it ends. For example, the sets in a game of volleyball, or the breaks in snooker.

We propose a method to extract information about the objects of interest in the game of snooker, so that it can be later used to detect events.

2.2. Color Based Tracking

Color based tracking [5, 7], are object tracking methods that use the colors of the objects as the main feature to track. This is specially relevant in the context of snooker, where the balls all look the same, with the exception of their color.

In this paper we don't only use the color to distinguish tracked objects, but we also train a neural network to know what class of color the ball is, this is important because most of the time the diference between a play being foul or not is closely related to the color of the balls.

Figure 1. Table processing. From top to bottom, left to right we have: Original image, Otsu's method, contour detection, largest contour, contour corners and homography.

3. Proposed Method and Results

3.1. Color Classification

In the proposed method we need some way to classify the colors of the balls, to that end, we train a small neural network with one input layer, one hidden layer, and one output layer. The input is the RGB value of the color, and the output the one of the eight possible color classes. The accuracy of the trained model was 90.32% after training for 500 epochs. However, in practice for snooker ball colors, we did not observe any misclassifications being made.

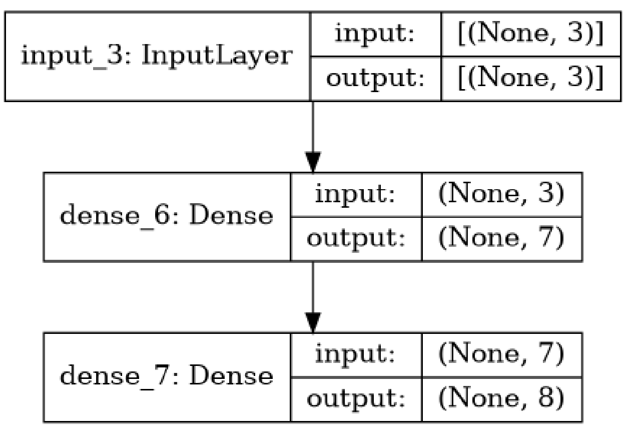

Figure 2. Model for color classification.

We used a already existing rgb color dataset[1], and modified it to our needs. Namely removing uneeded colors, and adding more colors to the dataset. To obtain new colors we trained the network using the available colors, and used it to classify the colors of the balls, and saved the results in a file. Then we manually filtered out bad results and fed the remaining to the network to get better accuracy.

3.2. Table Processing

We start the proposed method by detecting the table, and computing a homography for it. To do this we start by getting the first frame of the video, and using Otsu's method[4] to get the foreground objects, the most prevalent of which is the table. We then detect the contours of the resulting image. Since we know that the table is the largest foreground object, we can safely assume that the largest contour is the table itself. We then approximate a rectangle to the shape of the contour, and use that rectangle's corners to compute the homography. The results are shown in 1

Now that we remove the baize from the image. To do this we compute the histogram of the image, and select the most abundant color. We then filter out all the pixels that are in a small range around said color. The result is an image that only contains the balls, and the table frame.

3.3. Ball Detection

The next step is detecting the balls in the frame, now that we have removed the baize, all we need to do is detect the contours and threshold their areas so that we don't detect noise or the table frame as balls. We are left with either single balls, or groups of balls.

By doing this every frame, we're effectively tracking the balls, however we cannot yet distinguish between different balls.

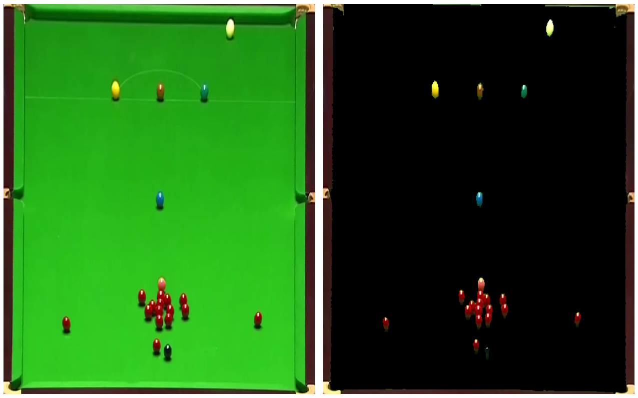

Figure 3. Baize removal method.

3.3.1 Ball Classification

Now that we have the ball contours, we classify them by color. To this end we do a kmeans clustering [3] with a single cluster, to get the predominant color of the ball. Then we feed the resulting center to the color classifier. And the result is the color class of the ball.

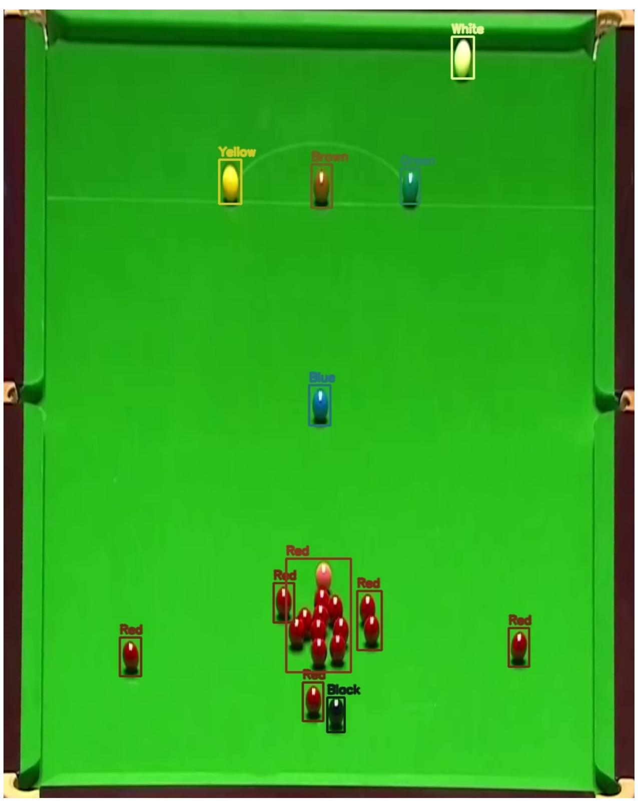

Figure 4. Detected balls with color classes.

3.4. Collision Detection

Another important piece of information that we need to extract from the footage are collisions. For the purposes of arbitrating we are only interested in the first collision of the cue ball.

Since collisions only occur with motion, it is unnecessary to check them for static balls. Thus in order to detect motion, the optical flow was first estimated by the means of the absolute difference between two consecutive frames. Taking the image generated by this difference, the contours present in it are taken and the Euclidean distance between all points of the contours is obtained. A collision is detected if the Euclidean distance between any two points on the contours is less than 60 pixels, a threshold obtained through several tests.

Furthermore, when a collision is detected its position and direction are estimated. To obtain the position, we take the first pixel belonging to the first contour, where the Euclidean distance to the second contour is smaller than the stipulated threshold. For the direction, we compute the difference between the centroid of the first contour and the centroid of the second contour and normalize the obtained vector.

We then use the collision location and direction to create a small circular mask around where the colided ball was. Then we apply this mask on the first frame, and use the ball detection and classification techniques to obtain the ball's color. We do this so that we can avoid classifying the color of multiple balls in one collision.

4. Conclusion and Future Works

We were able to extract most of the relevant information about a snooker shot. However for this method to be used to create an autonomous snooker referee we still need to extract some more information.

Firstly, we need to automatically split footage per shot, so that we can arbitrate each shot individually. One way to do this would be to track the movement of the cue ball, a shot begins when the cue ball starts moving and ends a few frames before the next shot begins. Also there is a performance problem, executing kmeans and using a neural network to classify the color of each ball every time is computationally expensive, this could be remediated by feeding less points to the kmeans clustering algorithm, and only trying to classify balls that participate in a shot. Other further optimizations might be necessary to improve the performance and accuracy of the method.

Furthermore, extracting information about a shot, and actually arbitrating, are two very different things. The title of the paper is "RuiBot: An Autonomous Snooker Referee", but we have not proposed an autonomous referee, just the building blocks necessary to create one. So the most signif-

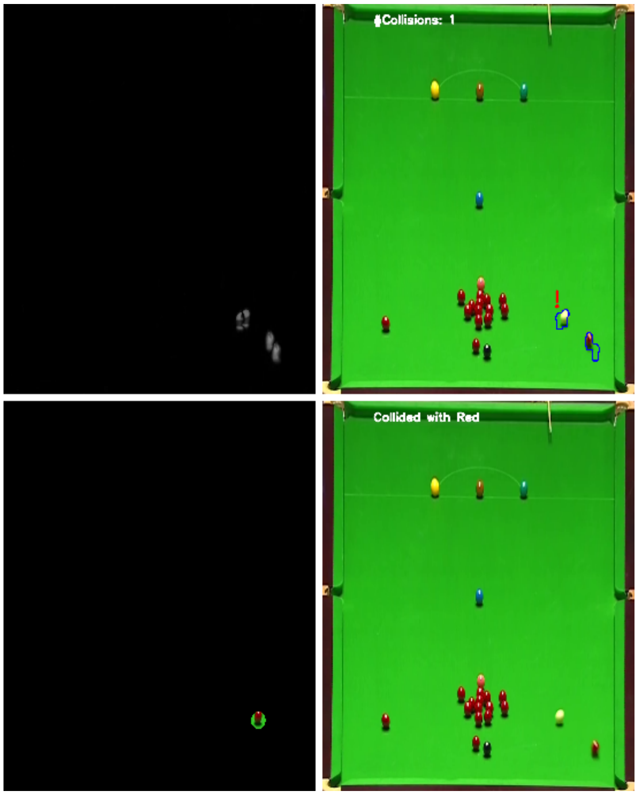

Figure 5. On the top left the optical flow, on the top right the detected collision, at the bottom left the circular mask for that collision, and at the bottom right the collided ball color classification.

icant future work would be actually using the information we have extracted to arbitrate shots, then breaks, then entire matches.

References

[1] Ajinkya Chavan. Color dataset. https://github.com/AjinkyaChavan9/RGB-Color-Classifier-with-Deep-Learning-using-Keras-and-Tensorflow/blob/master/Dataset/final_data.csv. Accessed: 2021-08-20.

[2] Rene Leveaux. Facilitating referee's decision making in sport via the application of technology. *Communications of the IBIMA*, 04 2010.

[3] S. Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28:129–137, 03 1982.

[4] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, Jan. 1979.

[5] Patrick Pérez, Carine Hue, Jaco Vermaak, and Michel Gangnet. Color-based probabilistic tracking. volume I, pages 661–675, 05 2002.

[6] Niall Rea, Rozenn Dahyot, and Anil Kokaram. Semantic event detection in sports through motion understanding. pages 88–97, 01 2004.

[7] Yongduek Seo, Sunghoon Choi, Hyunwoo Kim, and Ki-Sang Hong. Where are the ball and players? soccer game analysis with color-based tracking and image mosaick. In *Image Analysis and Processing*, pages 196–203. Springer Berlin Heidelberg, 1997.

[8] Dian W. Tjondronegoro and Yi-Ping Phoebe Chen. Knowledge-discounted event detection in sports video. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 40(5):1009–1024, Sept. 2010.

[9] Grigorios Tsagkatakis, Mustafa Jaber, and Panagiotis Tsakalides. Goal!! event detection in sports video. *Electronic Imaging*, 2017(16):15–20, Jan. 2017.