MULTIPLE BILLIARD BALLS TRACKING ON A POOL TABLE

DHILIP KUMAR AND ARIAN WU

COLORADO SCHOOL OF MINES

INTRODUCTION

The ability to track an object is of huge importance for various areas such as the military, commercial applications, monitoring [1], sports [2], and robotics [3]. In the sports area, especially in ball-based sports, there have been various approaches to try to track the ball for various purposes, mainly for obtaining multidimensional data [4], or decision making during games [5]. In the case of the game of pool, [4] has described some advantages regarding ball tracking such as the ball color and the small region that the pool table represents, and some disadvantages such as the small ball size and the varying velocities of the balls.

In this work, we are going to track multiple billiard balls being hit over a pool table. We will track their movements during a period of the game and overlay a line representing their movement.

PREVIOUS WORK

To this date, various efforts have been dedicated to the problem of ball tracking.

In the case of soccer, [2] tracked players and the soccer ball by using template matching between the actual frame and a template extracted in the previous frame, as well as implementing a Kalman filter. In [6], since the soccer ball tends to appear as white, they binarized the image to extract white pixels, followed by a morphological close operation, and finally evaluating the features of ball candidates such as the size of the object, the length-to-width ratio and the area of the object. Another strategy for fast ball detection has been implemented by [7] and [8] which consist of a "Coarse to Fine" strategy. This strategy is based on two steps: In the first step, the Coarse Step, any object whose features (area, contour, eccentricity, and form factor) don't resemble that of a ball are discarded. In the Fine Step, the gray template and circularity are further examined to determine the optimal region of the ball. Deep learning approaches have also been explored in [5], where they use a modified version of the VGG-M network, one of the smallest convolutional networks.

TECHNICAL APPROACH

A. EXPERIMENTAL SETUP:

On top of a pool table, 5 billiard balls were placed in different locations. A camera was placed over the table so that it can see the whole table. Four ArUco markers were placed at the corners of the table to help with the detection. For the experiment, the cue ball was hit by the cue stick and the recorded video was then processed.

B. ARUCO MARKER PLACEMENT:

In order to detect the corners of the pool table, 4 ArUco markers were placed in each of the corners. Then in the first frame of the recorded video, the ArUco markers were detected using OpenCV [9]. Then, knowing the dimensions of the pool table, an Homography Transform was found so that the ArUco markers would be transformed to the corners of the image, obtaining a warped image Iwarp.

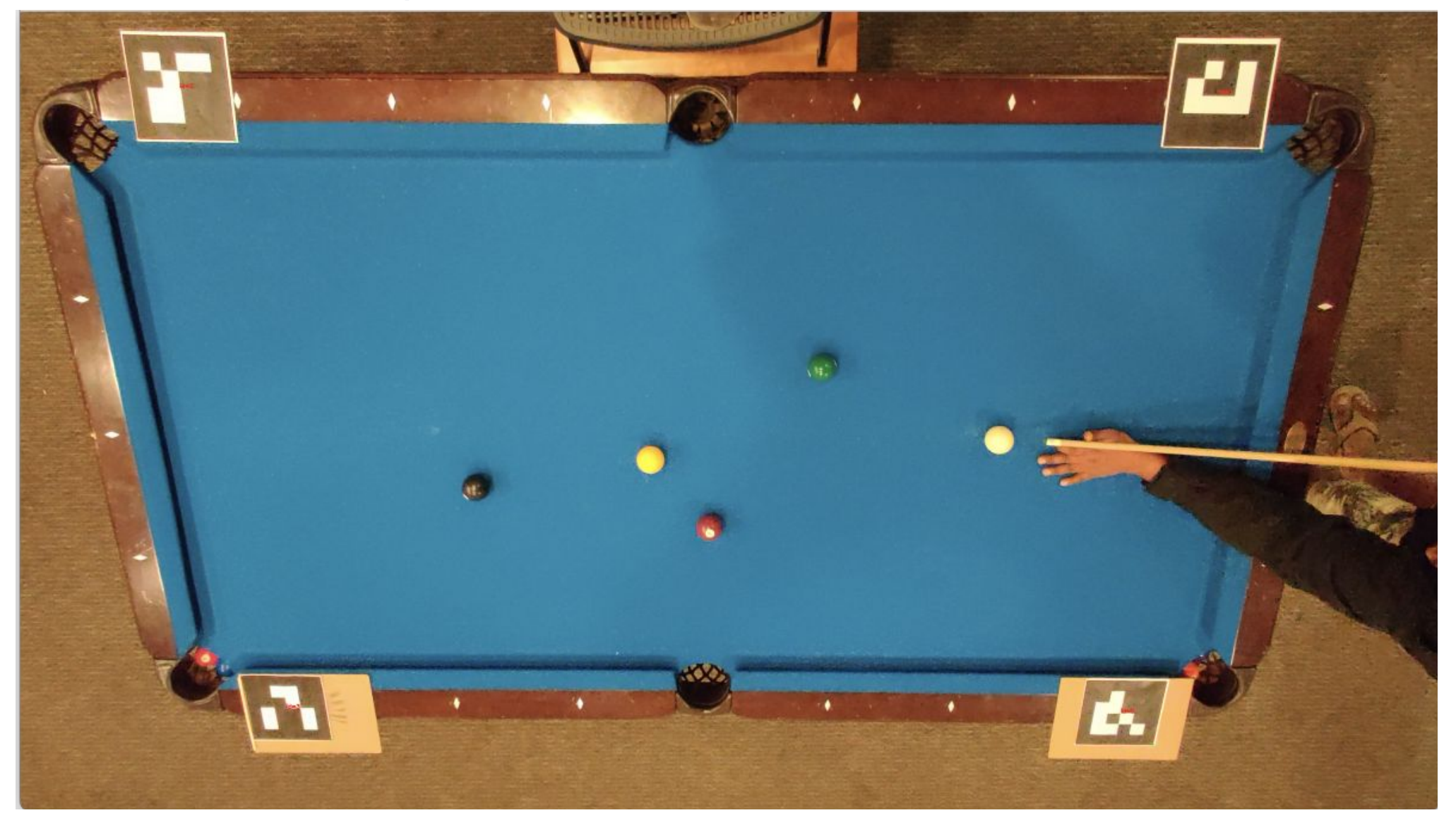
C. BILLIARD BALL DETECTION:

In order to perform the Ball detection, we first had to perform a Gaussian Blur to remove the noise from the video. Then, the image was converted from the RGB colorspace to the HSV colorspace, obtaining Ihsv . After converting the colorspace, Ihsv was thresholded using the range of color the table had present, in order to filter out the table and obtain an image with blobs representing the balls. Afterward, we perform morphological opening closing operations with an elliptical kernel in order to reduce any minor blobs that could be a part of the table and to make the blobs be solid. With that, we obtain a binary image Ibin.

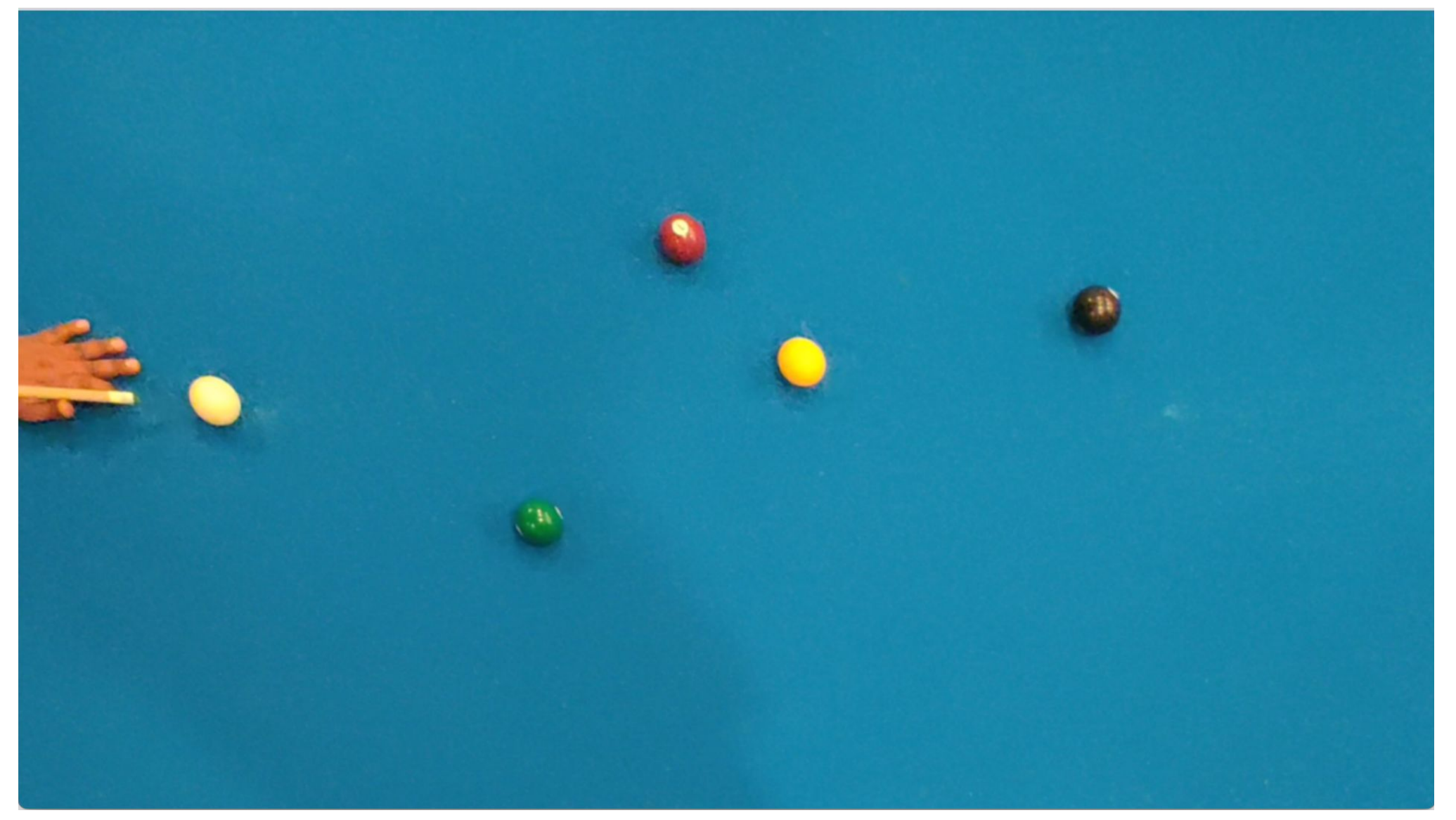
After obtaining Ibin, where the white pixels represent the foreground object, we find the individual contours. Then, we use the function minEnclosingcircle [9], choosing only contours with a radius between 60 and 100 and an area range of 8000 to 25000. With the obtained circles, apply the average color to the inner part of the circle, so that the circles are uniform in color. For tracking the balls, we first have to identify the balls in the next frame, in order to achieve that we use the average color of each ball. If the L2-norm between a ball and its previous frame is less than 40, we have found the ball. We then calculate the distance between the center of the ball and its last recorded position. If the distance is above 100 units we record the position to its corresponding list of positions. For visualization, we draw a line going from the first recorded point to the last.

RESULTS

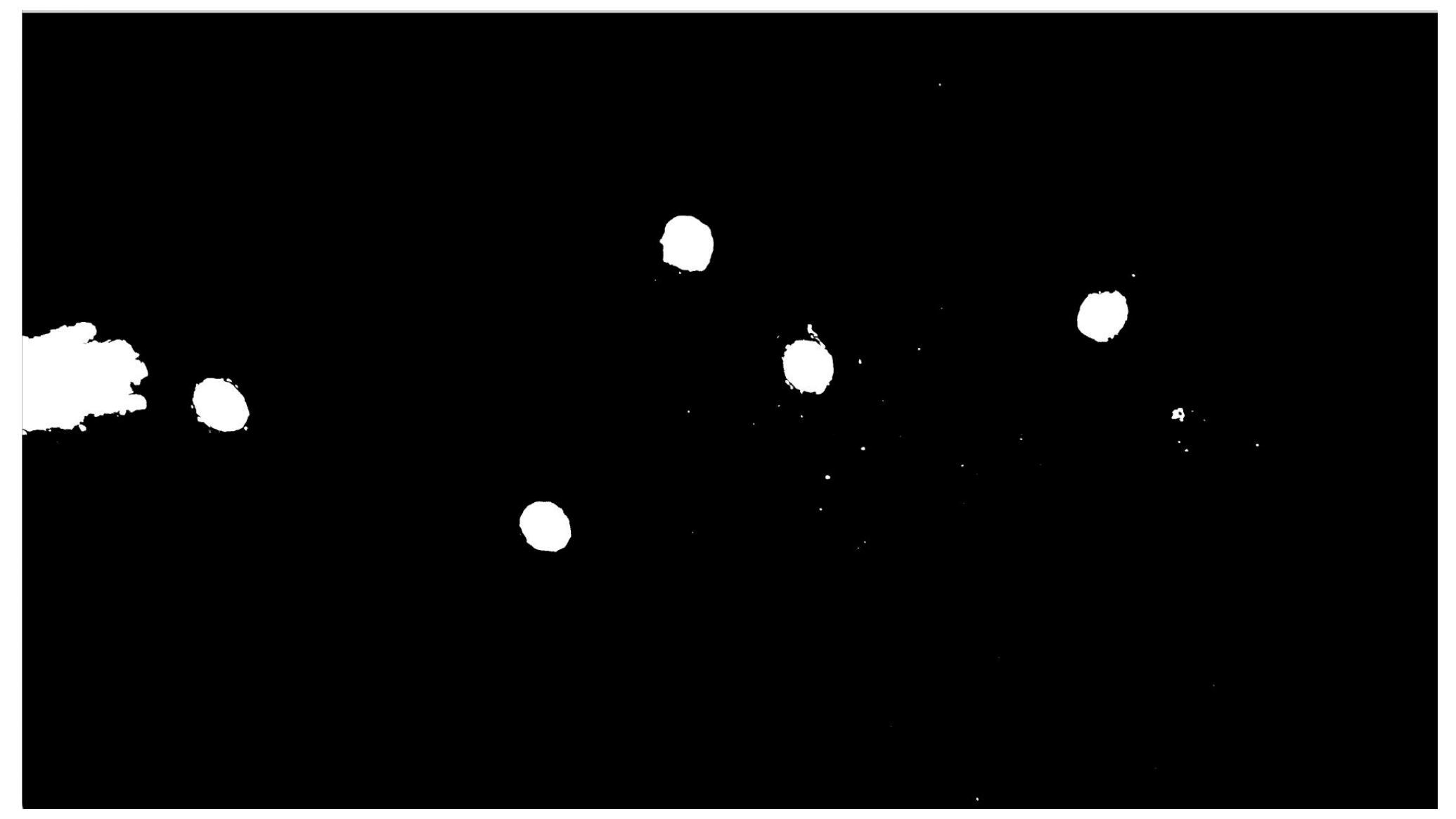
In the following images, each step taken for each frame will be shown. The image shown below is a frame taken from our test video. The ArUco markers can be seen placed near the corners of the pool table:



After detecting the ArUco markers, the image was warped. In the next figure, the frame can be seen warped and gaussian blur was added to the frame as well:

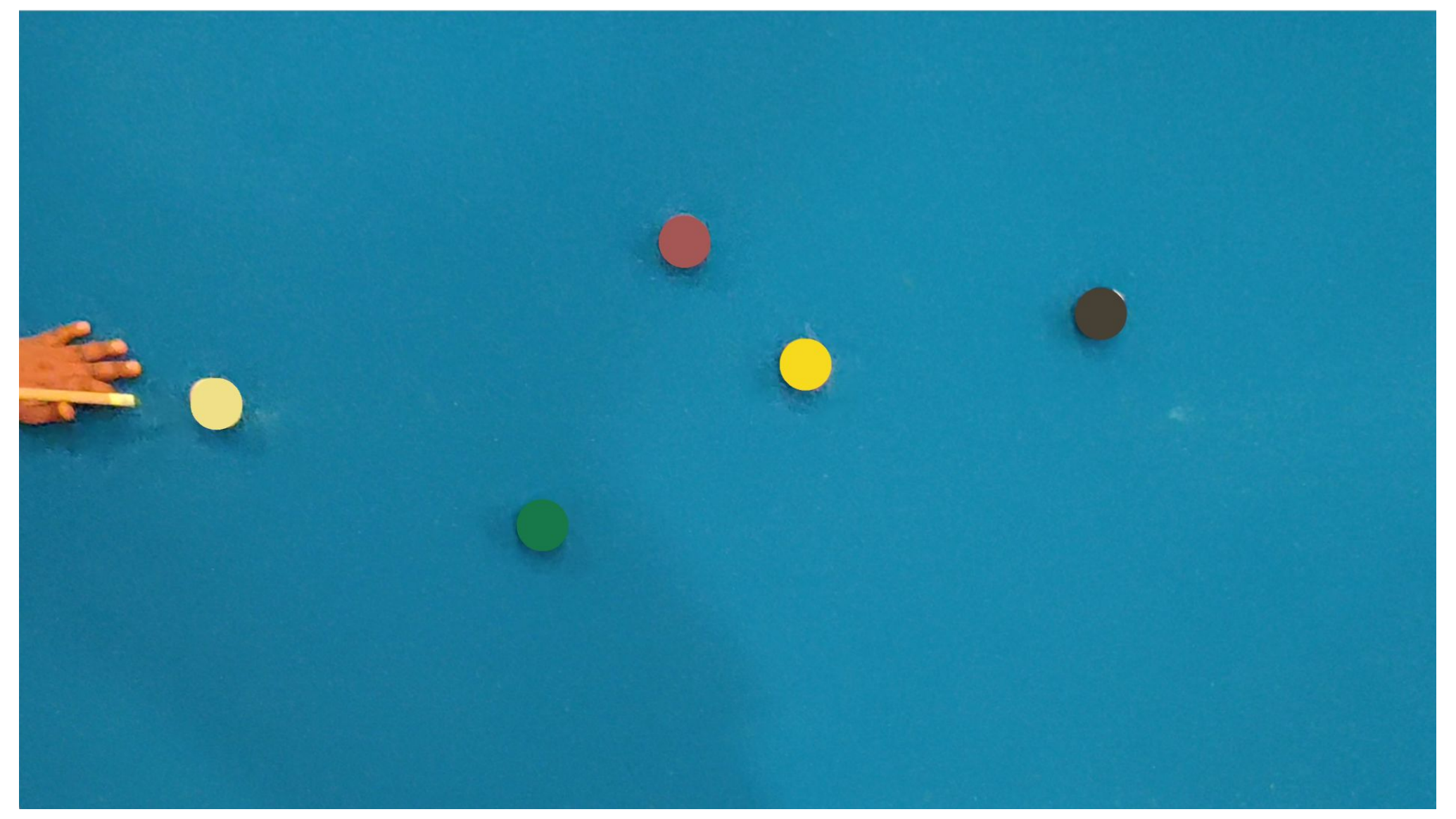


The next figure, is the frame after being transformed to HSV colorspace, thresholding the background (pool table) using a predefined range of colors, and performing a morphological opening followed by a morphological closing:

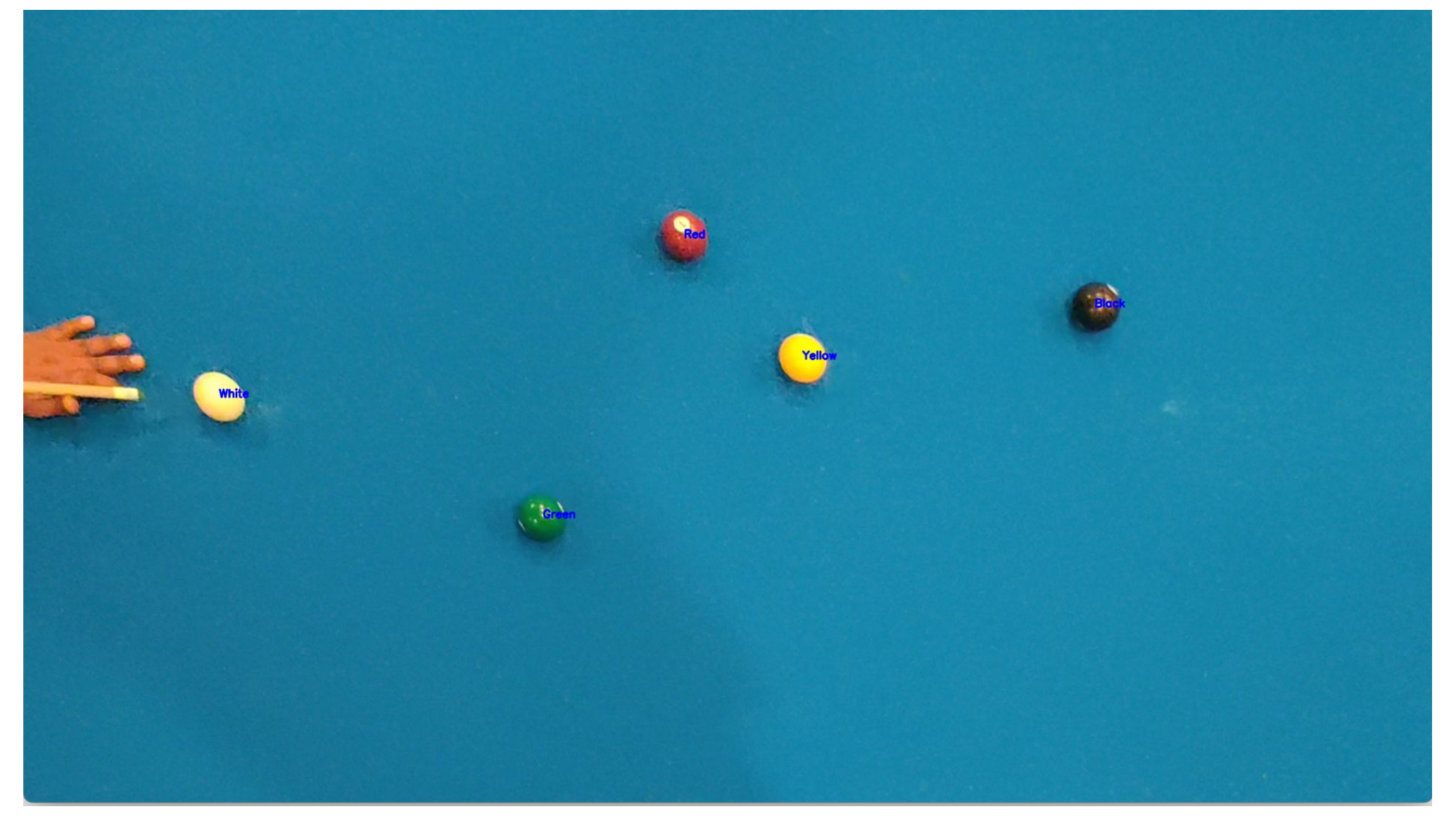


RESULTS

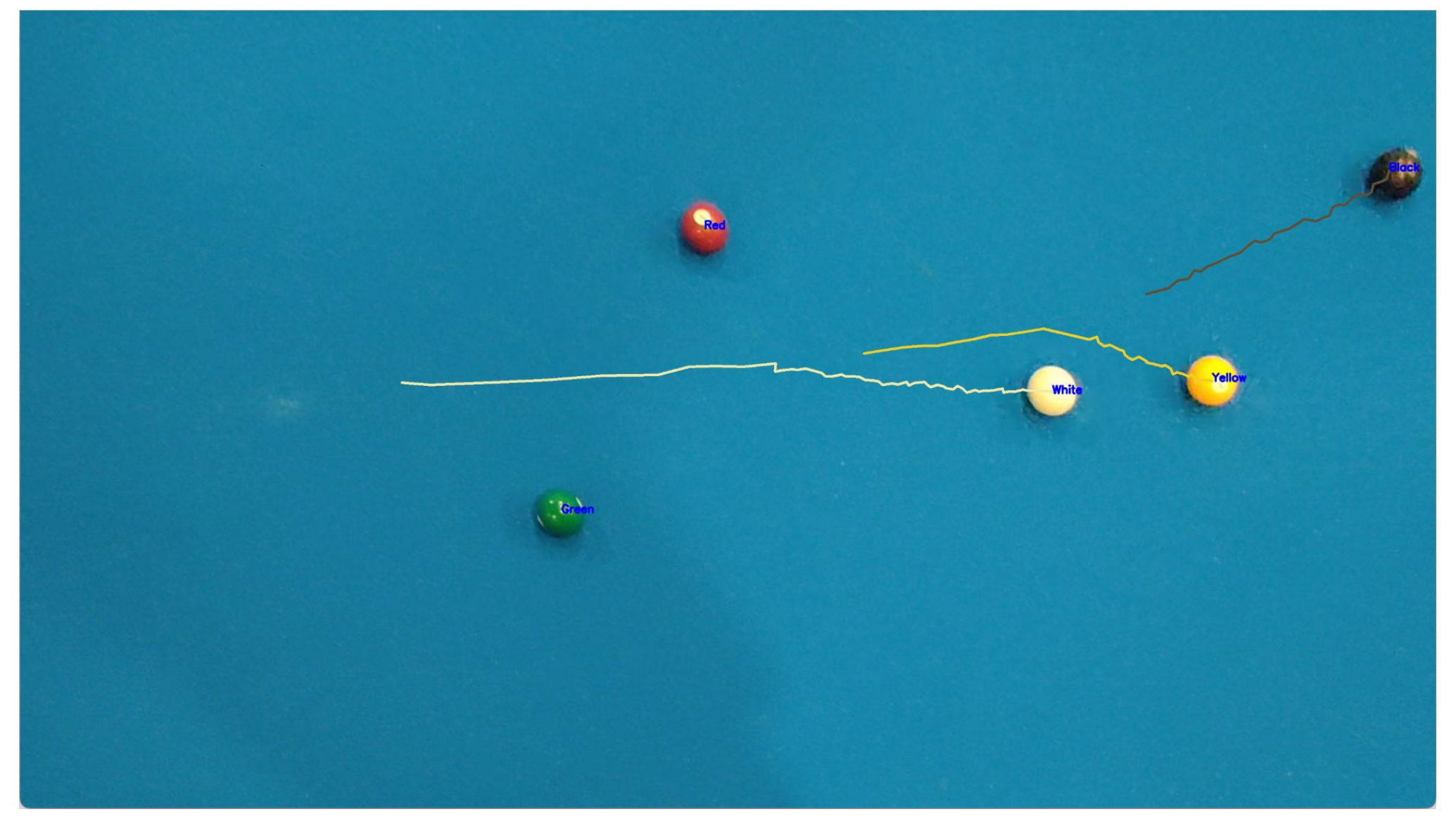
After identifying the foreground objects, the function minEnclosingCircles [9] was used by choosing only contours with a radius between 60 and 100 and an area range of 8000 to 25000. With this newfound circles, the average color within each circle was applied to the circle. The figure shown below is the image of the balls with the average color inside.



To identify each ball in the following frames, the balls closest in color are matched. Below is shown a frame with all balls identified per color:



Below is shown the final frame after the balls were tracked:



CONCLUSION

- We were able to successfully use the ArUco markers to determine the corners of the pool table
- We successfully isolated the contours bases in the HSV colorspace and identify circular figures on frame
- We successfully used morphological operations to clean the binarized image and obtain blobs that represented the foreground objects (the balls)
- We were able to identify each individual ball by determining the average color within its diameter and finding the closest ball in color in the next frame.
- We achieved ball tracking by recording the positions of each individual move for movements greater than 100 pxs

REFERENCES

[1] O. Ramos, M. Mirzaei, and F. Merienne, "Tracking in presence of total occlusion and size variation using mean shift and kalman filter," Science Arts & M´etiers, 12 2011.
[2] Y. Seo, S. Choi, H. Kim, and K.-S. Hong, "Where are the ball and players? soccer game analysis with color-based tracking and image mosaick," in Image Analysis and Processing, A. Del Bimbo, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1997, pp. 196–203.
[3] H.-I. Lin and Y.-C. Huang, "Ball trajectory tracking and prediction for a ping-pong robot," in 2019 9th International Conference on Information Science and Technology (ICIST), 2019, pp. 222–227.
[4] P. R. Kamble, A. G. Keskar, and K. M. Bhurchandi, "Ball tracking in sports: A survey," Artificial Intelligence Review, vol. 52, no. 3, p. 1655–1705, 2017.
[5] P. Kamble, A. Keskar, and K. Bhurchandi, "A deep learning ball tracking system in soccer videos," Opto-Electronics Review, vol. 27, no. 1, pp. 58–69, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S123034021830146X
[6] D. Liang, Y. Liu, Q. Huang, and W. Gao, "A scheme for ball detection and tracking in broadcast soccer video," in Advances in Multimedia Information Processing - PCM 2005, Y.-S. Ho and H. J. Kim, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 864–875.
[7] X.-F. Tong, H.-Q. Lu, and Q.-S. Liu, "An effective and fast soccer ball detection and tracking method," in Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004., vol. 4, 2004, pp. 795–798 Vol.4.
[8] X. Yu, C. Xu, Q. Tian, and H. W. Leong, "A ball tracking framework for broadcast soccer video," in 2003 International Conference on Multimedia and Expo. ICME '03. Proceedings (Cat. No.03TH8698), vol. 2, 2003, pp. II–273.
[9] G. Bradski, "The OpenCV Library," Dr. Dobb's Journal of Software Tools, 2000