UNIVERSITÀ DEGLI STUDI DI PADOVA

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

# Computer Vision Project

## Sport Video Analysis for Billiard Matches

Federico Pivotto 2121720, Fabrizio Genilotti 2119281, Leonardo Egidati 2106370

**Pogi Team**

## 1 Introduction

In the competitive billiards sector it is common to use automated artificial vision systems that allows to monitor the status of the match and provide a high-level view of the playing field. These systems allow players to study strategies for improving their performance in game.

To this purpose, the proposed computer vision system allows to analyze a video dataset and produce for each one additional information regarding the state of the game.

The system consists in an executable that generates the following outputs for each video:

1. First and last frame of the video;
2. First and last frame with border detection of the playing field;
3. First and last frame with segmentation of the playing field;
4. First and last frame given by the union of 2 and 3;
5. First and last frame with mask segmentation of both the playing field and the background;
6. First and last frame with ball detection and classification;
7. First and last frame with 2D top-view visualization map;
8. Text file with a list of ball bounding boxes, one for each row, detected in first and last frame;
9. Video with 2D-top-view visualization map;
10. Metrics: mAP and mIoU assessed on first and last frame.

To generate these results, the system is based on five components: border detection, ball detection and classification, playing field segmentation, 2D top-view visualization map and tracking, and finally metrics.

Two types of metrics are used to evaluate the system performance: mean Average Precision (mAP) and mean Intersection over Union (mIoU). These are measured on the test dataset containing 10 videos, each with related ground-truth annotations.

## 2 Border Detection

The first step of this computer vision system consists in detecting the borders of the playing field. Given a video frame, to complete the task, the adopted approach consists in segmenting the table in the HSV color space. More precisely, the color segmentation produces a binary mask, which is enhanced by applying in sequence two morphological operations: dilation followed by erosion with rectangular structuring element. Once the mask is computed, Canny edge-detection is applied on it to find the candidate lines for each border of the table. Then, the candidate lines are grouped in four groups by similarity of their (rho, slope) values, which are used to perform lines suppression, leaving only the four final borders of the billiard playing field.

In addition to the borders detection, this approach exploits the selected borders to compute the corners location by finding their intersections.

# 3 Ball Detection and Classification

The second step of this computer vision system consists in detecting the location of the balls and then classifying each according to the four classes:

- 1: white "cue ball";
- 2: black "8-ball";
- 3: ball with solid color;
- 4: ball with stripes.

To complete the task, the adopted approach consists in pre-processing the given video frame in BGR color space applying in sequence the following filters: median filter for smoothing, difference with respect to the Gaussian blurred frame for sharpening, and bilateral filter for preserving edges retaining color information. The resulting frame is converted in HSV color space for generating the binary mask according to thresholds experimented and selected using trackbars. Then, to enhance the mask for highlighting the balls, a combination of morphological operators with rectangular and elliptic structuring elements is applied to it.

Once the mask is prepared, the Hough circle transform is applied to detect circular shapes in order to localize billiard balls. The collected circles are filtered according to a ordered sequence of suppression techniques: suppression of circles too close to billiard holes, computation of mean (center, radius) circles of really close ones, suppression of small and big circles, suppression of circles having non-convex shape in the mask (i.e. with center of 0 pixel intensity). Finally, the radius of the circles is normalized considering the median radius.

## 3.1 Ball Classification

After converting the obtained circles into bounding boxes, the task of ball classification is performed. As first step, we compute for each ball the ball's gradient with respect to the grayscale pre-processed frame, and the white and black ratios with respect to the BGR pre-processed frame. More precisely, the gradient is used to compute the gradient count, namely the number of pixels inside each ball region with magnitude greater than a certain threshold. The reason why it is considered the gradient count is that stripes balls should have a greater value than solid ones.

To detect the white "cue ball" and black "8-ball", first the balls with highest and second-highest white and black ratio, respectively, are identified. For each of the two colors the highest and second-highest balls are compared: if the ratios difference is negligible and the gradient count of the second-highest ball is less than the highest one, then the second-highest ball is classified as black/white ball, otherwise the highest one.

Finally, to distinguish stripes and solid color balls except the selected white "cue ball" and black "8-ball", white ratios and gradient counts are considered. A candidate ball is classified as stripes ball if its white ratio is greater than a increased white ratio threshold, since the presence of a large white region likely imply the stripes ball class. If it is not the case but it still has a greater white ratio than the normal white ratio threshold and the gradient count is greater than a fixed gradient threshold, then it still imply the stripes ball class. If none of these two cases is satisfied, then the ball is classified as solid ball.

# 4 Playing Field Segmentation

The third step of this computer vision system consists in semantically segmenting the playing field with balls. Given a video frame, to complete the task, the adopted approach consists in exploiting the detected borders to identify the region corresponding to the playing field and fill it with green color. Then, the previously localized and classified balls are drawn on the resulting frame distinguishing in color the four ball classes.

Starting from a zeroed mask, the system also generates in the same way the segmentation mask, this time associating to each class the gray level corresponding exactly to its class ID.

# 5 2D Top-View Visualization Map and Ball Tracking

The fourth step is to generate the 2D map-view of the billiard table, which serves as a compact representation useful for tracking of the balls' trajectories during a match. This task can be assessed by using the borders, the corners' locations on the playing field and finally the balls' bounding boxes. First of all, to correctly generate the map-view based on the table orientation, the perspective distortion caused by the camera angle is checked. The presence of distortion is determined by checking the sum of the borders' slopes: if the sum of slopes is close to zero, there is likely some non-negligible distortion in the perspective. As a simplifying assumption, the presence of distortion is directly correlated with the table orientation. Once the table orientation is determined, the four corners of the playing field are used to calculate the homography matrix, which maps the field itself to the map-view. The mapping obtained by means of the homography matrix is then used to compute the center of each bounding box when warped to the map-view. This calculation can be done for each bounding box center by multiplying the homography matrix by the center's homogeneous coordinates. For each bounding box, the warped center coordinates are then used as ball center in the map-view to accurately represent and draw the corresponding ball.

### 5.1 Ball Tracking

To start tracking the balls' position frame by frame of each video in the dataset, it is associated a Channel and Spatial Reliability Tracking (CSRT) tracker to each localized and classified ball stored in the annotations previously predicted by the system. The reason why this type of tracker was chosen is given by its accuracy: the trade-off is a slow computational speed.

In order to make the tracking more reliable and robust with respect to the lost of targets, augmented ball bounding boxes by 150% are considered. Once the ball trackers are sat, they are updated for each frame next to the first one until the last video frame. All tracked ball positions are marked in the the 2D map-view visualization map to keep a real-time history of the billiard match going on.

## 6 Metrics

The last step of this computer vision system consists in evaluating the balls localization with respect to the four ball classes, and the semantic segmentation with respect to the background, the playing field and the four ball classes, for a given video frame. This evaluation is based on the comparison with the ground truth provided within the dataset through annotations.

The balls localization is evaluated through the mean Average Precision (mAP) which is given by the arithmetic mean of the Average Precision (AP), computed for each ball class, over the four ball classes. To compute the AP of each ball class, the corresponding balls in the ground truth and predicted annotations are matched, accepting ball matches with highest IoU greater or equal to IoU threshold 0.5. Each predicted ball involved in a match increases by one the number of True Positive (TP), otherwise the number of False Positive (FP). After sorting in descending order the TP and FP according to the balls confidence, assumed to be 1 for all predicted balls, the cumulative precision and recall are computed. Finally the AP is computed as the division of the sum of interpolated cumulative precision obtained according to PASCAL VOC 11 point interpolation method, by 11.

On the other side, the semantic segmentation is evaluated through the mean Intersection over Union (mIoU) which is given by the arithmetic mean of the class IoU, over the background, playing field, and four ball regions. To compute each class IoU, the ground truth and predicted segmentation masks of the given video frame are compared with respect to the gray level regions corresponding to the class ID. In both masks, these regions as seen as one, so that the IoU for the entire class can be computed as the ratio between the intersection area and the union area.

Both metrics, mAP and mIoU, are computed according to the two online resources provided in the assignment.

## 7 Conclusions

Sport video analysis for billiard matches can be successfully performed by computer vision systems. In the specific case of this computer vision system, it is possible to gather interesting information from the analyzed videos, such as ball detection and classification, and also map-view representation of the current match in a compact way. However, at the current state the system might lack a robust detection and classification of the balls in particular lighting conditions and in presence of occlusions. Future improvements could focus on enhancing ball localization by fine-tuning the bounding box dimensions and center, and also on the classification process by exploiting more effectively the gradient and the color information.

### 7.1 Working Hours and Member Ideas

The members of the project have spent the following number of hours to complete the computer vision system:

| **Members** | **Time spent** $[hour]$ |
|---|---|
| Federico Pivotto | 64.5 |
| Fabrizio Genilotti | 50 |
| Leonardo Egidati | 30.5 |
| Team (together) | 35.5 |
| **Total** | 180.5 |

Table 1: Individual and team working hours

In the following sections, we present the ideas contributed by each member of the team.

#### 7.1.1 Federico's Contributions

**System workflow:**

- General working logic of the system.

**Ball Detection:**

- Generation and enhancement of the playing binary mask;
- Setup of the Hough circle transform on the playing field mask.

**Ball Classification:**

- Calculation of white and black ratios of balls;
- Classification process exploiting ball gradient.

**2D Top-View Visualization Map:**

- Drawing of balls and playing field on the 2D top-view visualization map.

**Ball Tracking:**

- Use of CSRT tracker and drawing of ball trajectories on the 2D top-view visualization map.

### 7.1.2 Fabrizio's Contributions

**System workflow:**

- Integration of border detection and 2D top-view map in the working logic of the system.

**Border Detection:**

- Use of HSV mask with Hough line transform;
- Suppression of similar lines;
- Corner computation.

**Ball Classification:**

- Idea of using gradient in the ball classification.

**2D Top-View Visualization Map:**

- Use of homography matrix for mapping the playing field;
- Checking perspective distortion;
- Mapping bounding box centers in the 2D top-view visualization map.

### 7.1.3 Leonardo's Contributions

**System workflow:**

- Integration of field segmentation and metrics in the working logic of the system.

**Playing Field Segmentation:**

- Generation of frames with a segmented playing field.

**Metrics:**

- General logic for computing all the metrics.

# 8 Results

In order to assess the computer vision system performances, here are reported the resulting frames and the relative metrics values for each video in the dataset.

| | **First frame** | | **Last frame** | |
|---|---|---|---|---|
| **Game clip** | mAP [%] | mIoU [%] | mAP [%] | mIoU [%] |
| game1_clip1 | 84.090909 | 65.384192 | 90.909091 | 72.989396 |
| game1_clip2 | 88.636364 | 77.855979 | 86.363636 | 70.154443 |
| game1_clip3 | 90.909091 | 72.279336 | 90.909091 | 74.641990 |
| game1_clip4 | 84.090909 | 73.594513 | 50.000000 | 57.628080 |
| game2_clip1 | 79.545455 | 72.924810 | 86.363636 | 70.540783 |
| game2_clip2 | 29.545455 | 40.833224 | 54.545455 | 53.155875 |
| game3_clip1 | 77.272727 | 65.886124 | 88.636364 | 72.334454 |
| game3_clip2 | 79.545455 | 70.932385 | 72.727273 | 68.246856 |
| game4_clip1 | 59.090909 | 59.962848 | 63.636364 | 62.259338 |
| game4_clip2 | 54.545455 | 58.555372 | 79.545455 | 60.517731 |

TABLE 2: mAP and mIoU metrics computed for each dataset game clip

## 8.1 First frame of each video

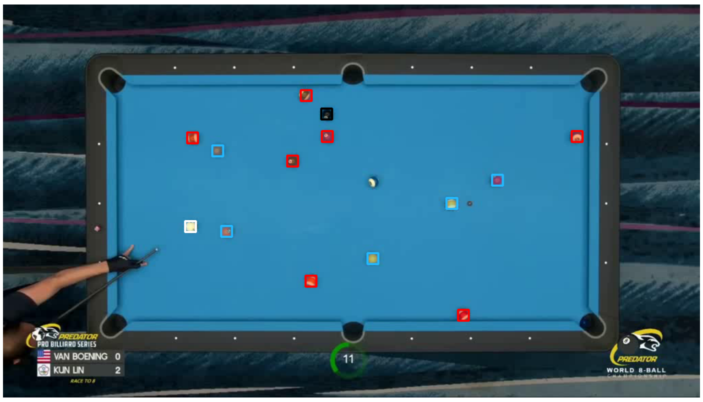


FIGURE 1: Object detection

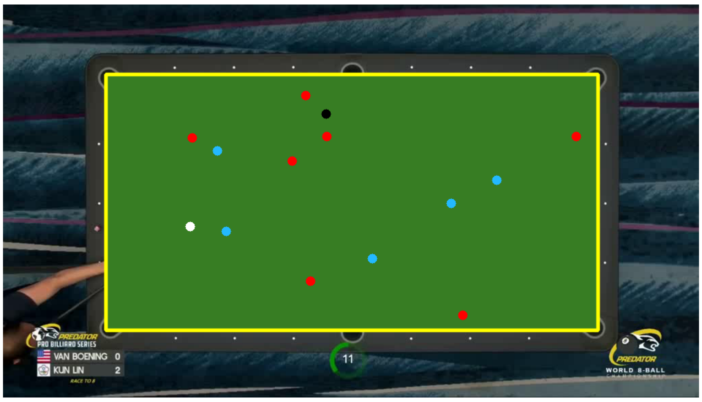


FIGURE 2: Playing field border detection and segmentation

FIGURE 3: Game 1, clip 1, first frame: $mAP = 84.090909\%$, $mIoU = 65.384192\%$


FIGURE 4: Object detection

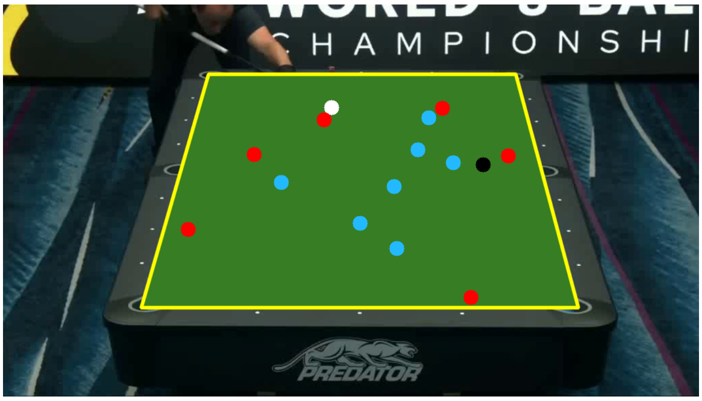


FIGURE 5: Playing field border detection and segmentation

FIGURE 6: Game 1, clip 2, first frame: $mAP = 88.636364\%$, $mIoU = 77.855979\%$

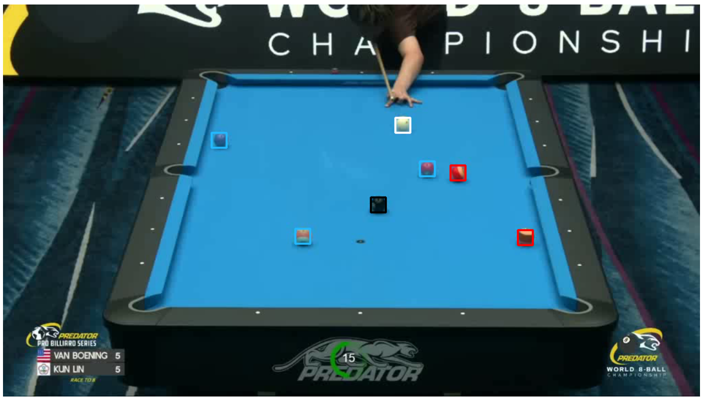


FIGURE 7: Object detection

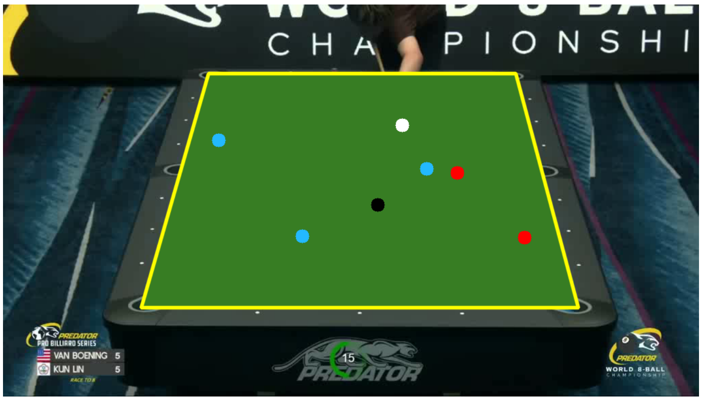


FIGURE 8: Playing field border detection and segmentation

FIGURE 9: Game 1, clip 3, first frame: $mAP = 90.909091\%$, $mIoU = 72.279336\%$

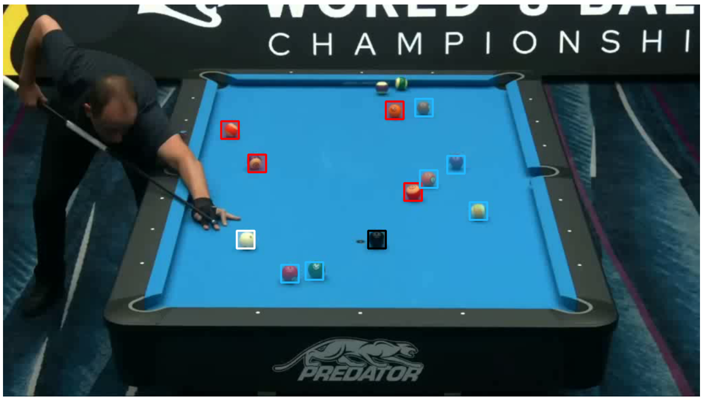


FIGURE 10: Object detection

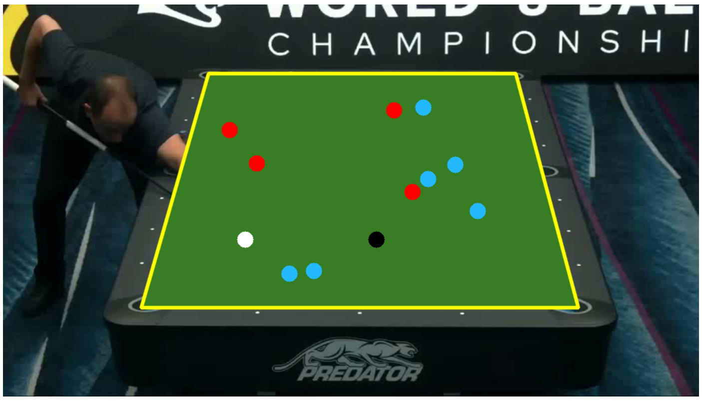


FIGURE 11: Playing field border detection and segmentation

FIGURE 12: Game 1, clip 4, first frame: $mAP = 84.090909\%$, $mIoU = 73.594513\%$

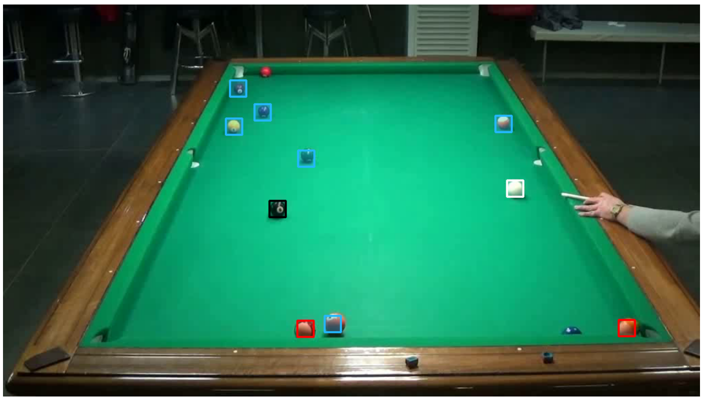

FIGURE 13: Object detection

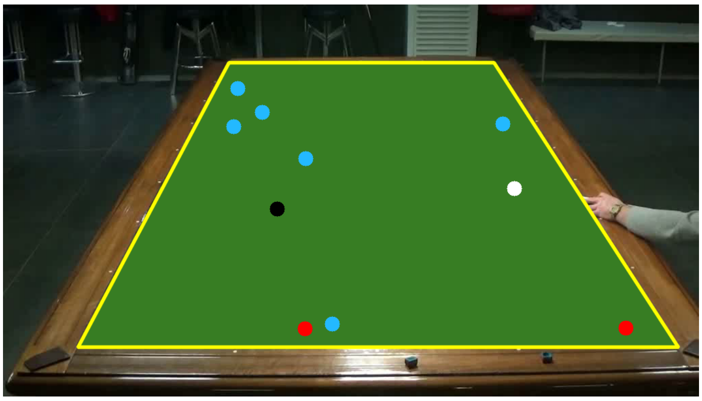

FIGURE 14: Playing field border detection and segmentation

FIGURE 15: Game 2, clip 1, first frame: $mAP = 79.545455\%$, $mIoU = 72.924810\%$

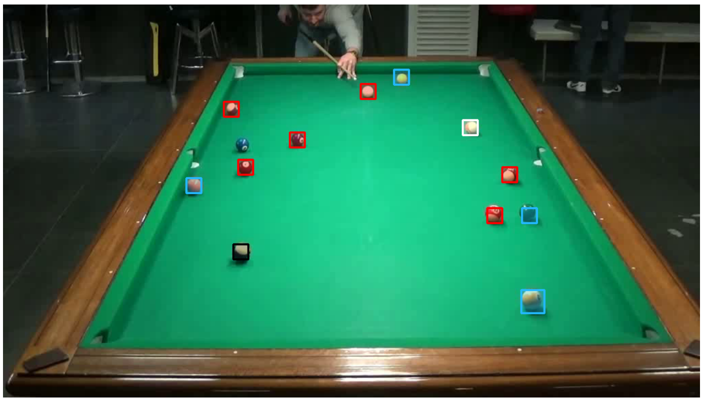

FIGURE 16: Object detection

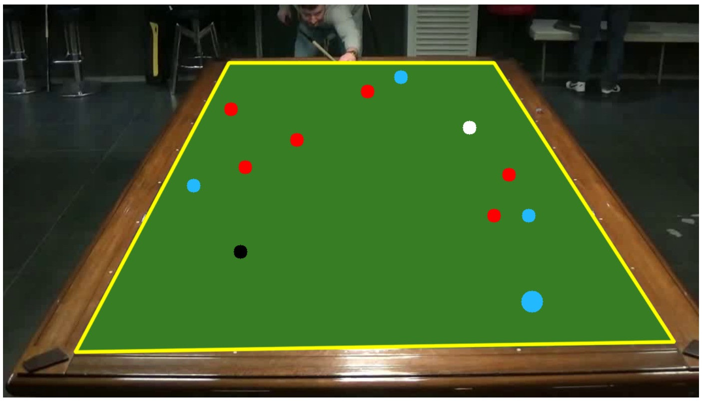

FIGURE 17: Playing field border detection and segmentation

FIGURE 18: Game 2, clip 2, first frame: $mAP = 29.545455\%$, $mIoU = 40.833224\%$


FIGURE 19: Object detection

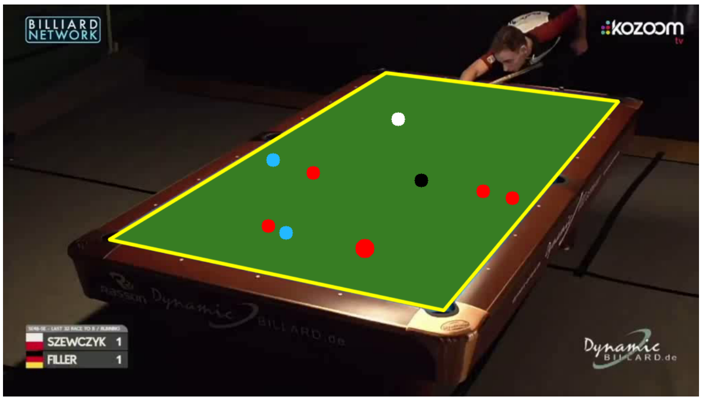


FIGURE 20: Playing field border detection and segmentation

FIGURE 21: Game 3, clip 1, first frame: $mAP = 77.272727\%$, $mIoU = 65.886124\%$

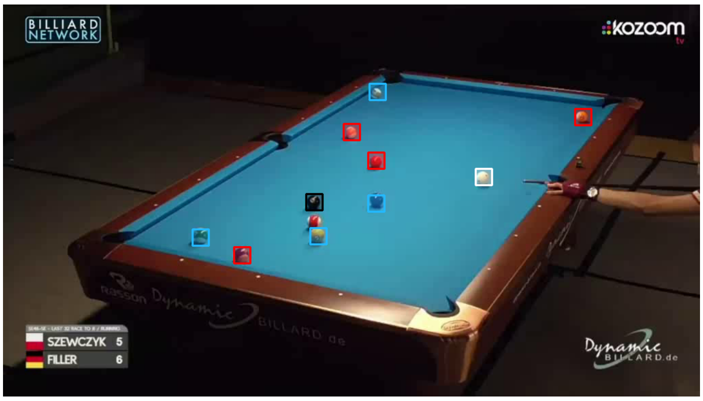


FIGURE 22: Object detection

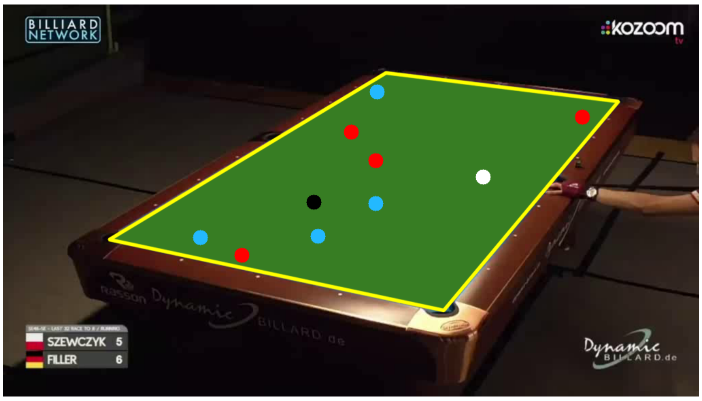


FIGURE 23: Playing field border detection and segmentation

FIGURE 24: Game 3, clip 2, first frame: $mAP = 79.545455\%$, $mIoU = 70.932385\%$

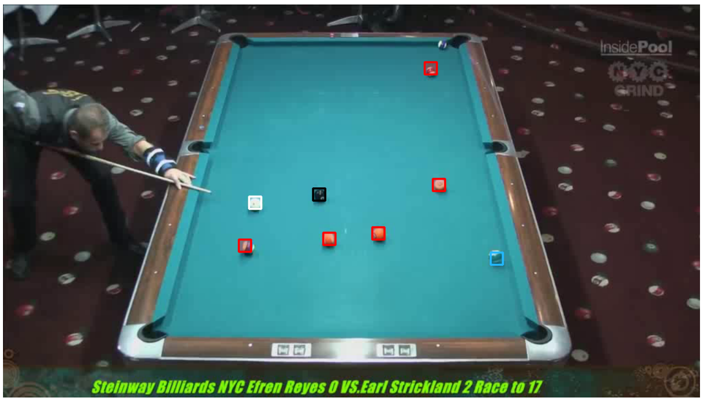


FIGURE 25: Object detection


FIGURE 26: Playing field border detection and segmentation

FIGURE 27: Game 4, clip 1, first frame: $mAP = 59.090909\%$, $mIoU = 59.962848\%$

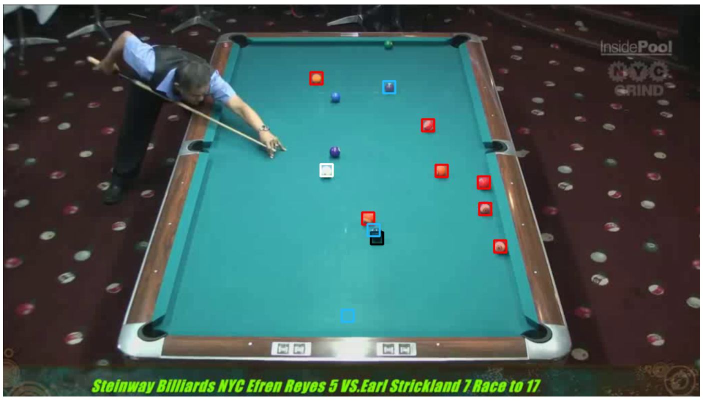


FIGURE 28: Object detection


FIGURE 29: Playing field border detection and segmentation

FIGURE 30: Game 4, clip 2, first frame: $mAP = 54.545455\%$, $mIoU = 58.555372\%$

### 8.2 Last frame of each video

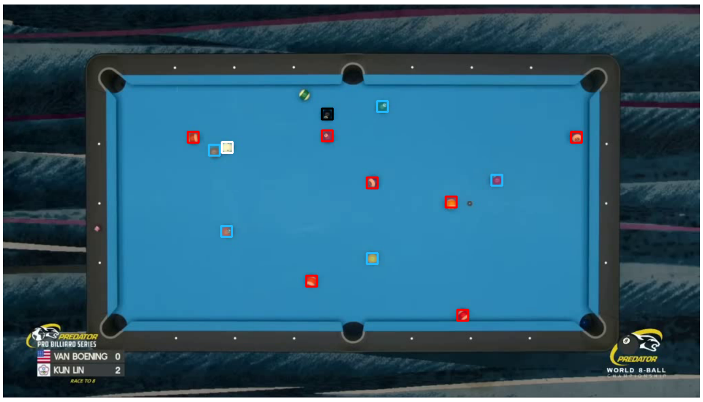


FIGURE 31: Object detection

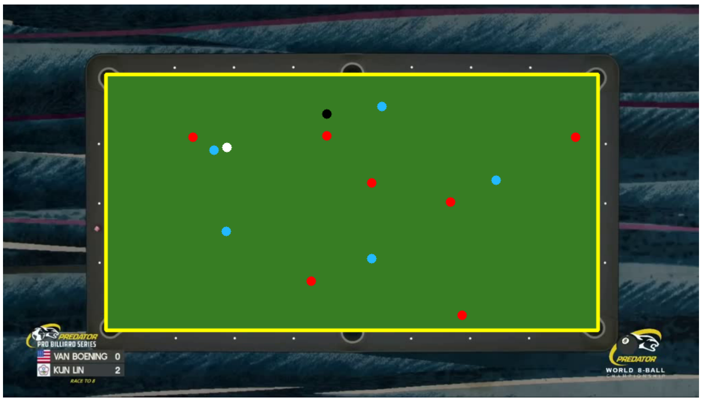


FIGURE 32: Playing field border detection and segmentation

FIGURE 33: Game 1, clip 1, last frame: $mAP = 90.909091\%$, $mIoU = 72.989396\%$

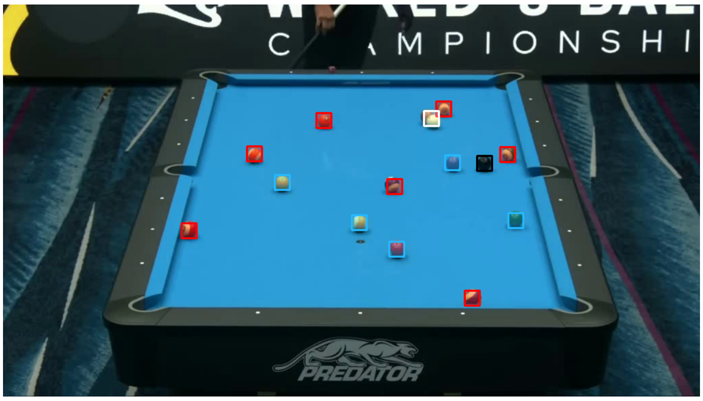


FIGURE 34: Object detection

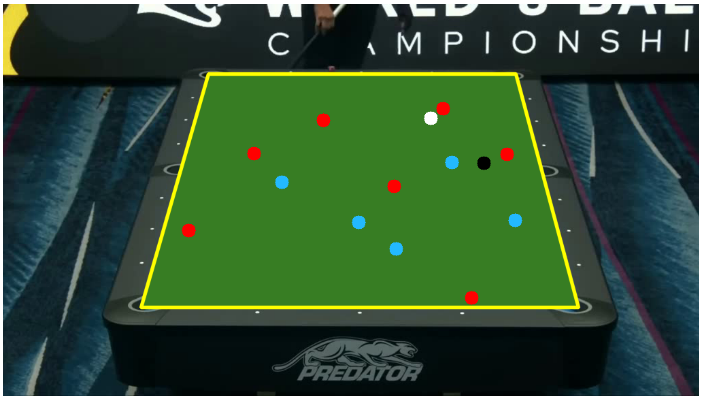


FIGURE 35: Playing field border detection and segmentation

FIGURE 36: Game 1, clip 2, last frame: $mAP = 86.363636\%$, $mIoU = 70.154443\%$


FIGURE 37: Object detection

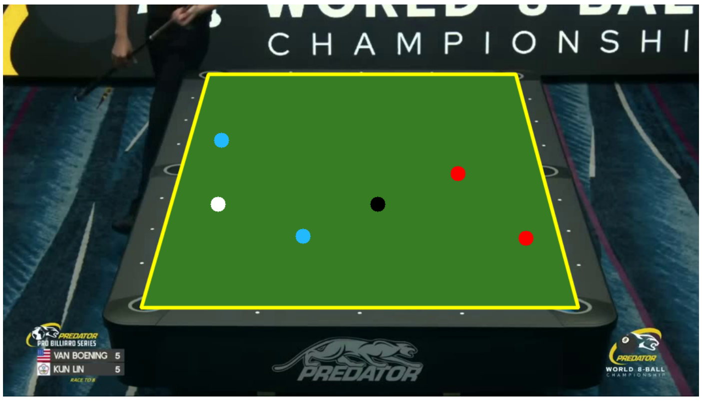


FIGURE 38: Playing field border detection and segmentation

FIGURE 39: Game 1, clip 3, last frame: $mAP = 90.909091\%$, $mIoU = 74.641990\%$


FIGURE 40: Object detection

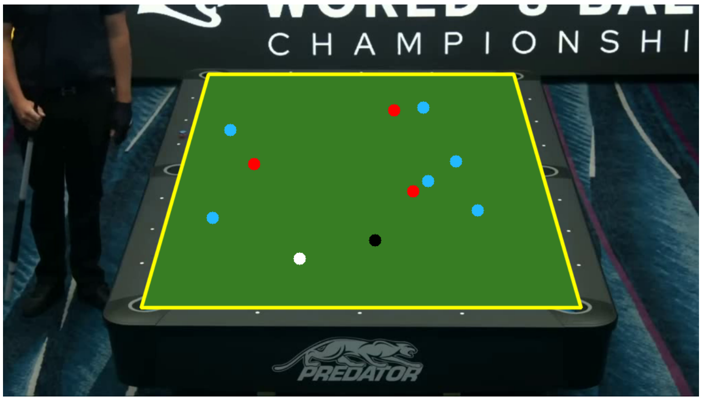


FIGURE 41: Playing field border detection and segmentation

FIGURE 42: Game 1, clip 4, last frame: $mAP = 50.000000\%$, $mIoU = 57.628080\%$

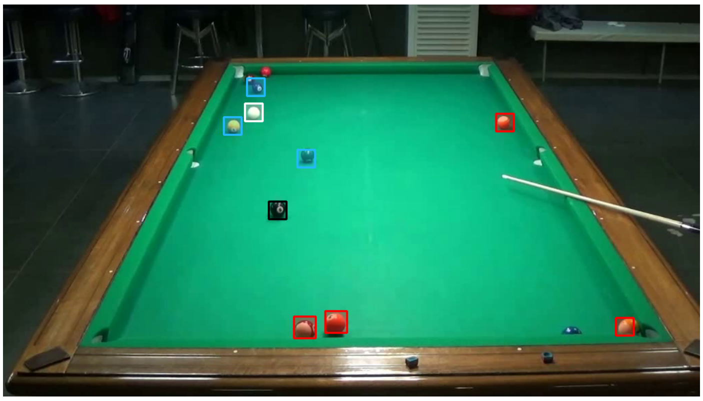

FIGURE 43: Object detection

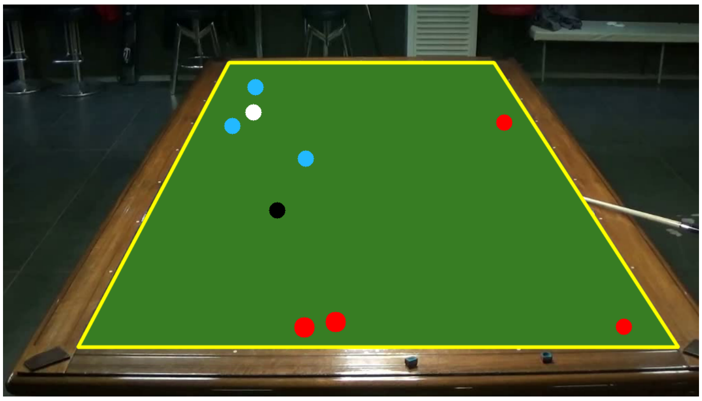

FIGURE 44: Playing field border detection and segmentation

FIGURE 45: Game 2, clip 1, last frame: $mAP = 86.363636\%$, $mIoU = 70.540783\%$

FIGURE 46: Object detection

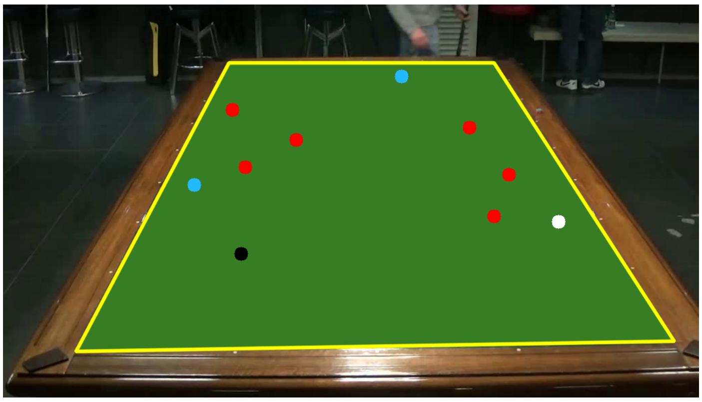

FIGURE 47: Playing field border detection and segmentation

FIGURE 48: Game 2, clip 2, last frame: $mAP = 54.545455\%$, $mIoU = 53.155875\%$

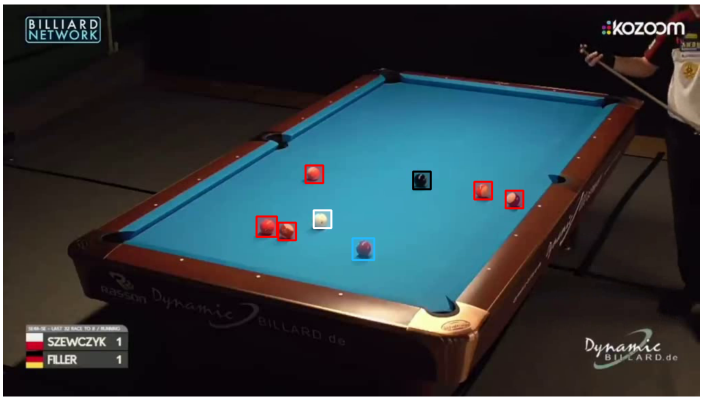


FIGURE 49: Object detection

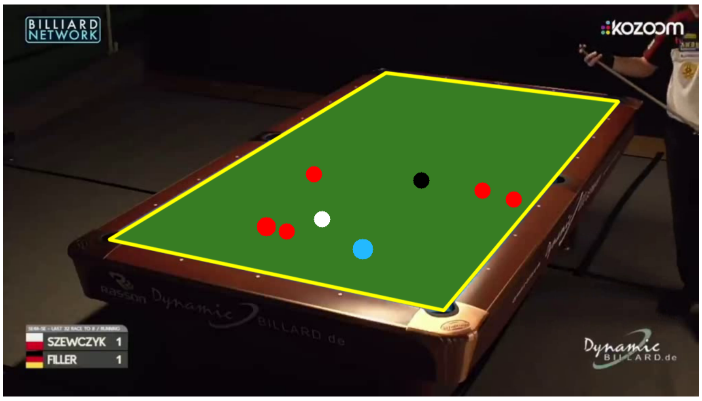


FIGURE 50: Playing field border detection and segmentation

FIGURE 51: Game 3, clip 1, last frame: $mAP = 88.636364\%$, $mIoU = 72.334454\%$

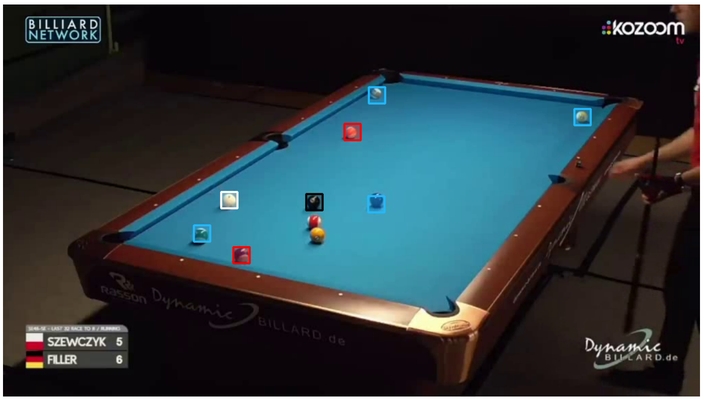


FIGURE 52: Object detection

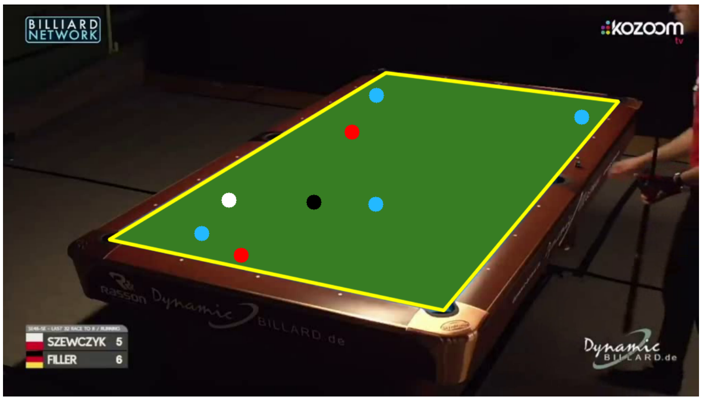


FIGURE 53: Playing field border detection and segmentation

FIGURE 54: Game 3, clip 2, last frame: $mAP = 72.727273\%$, $mIoU = 68.246856\%$

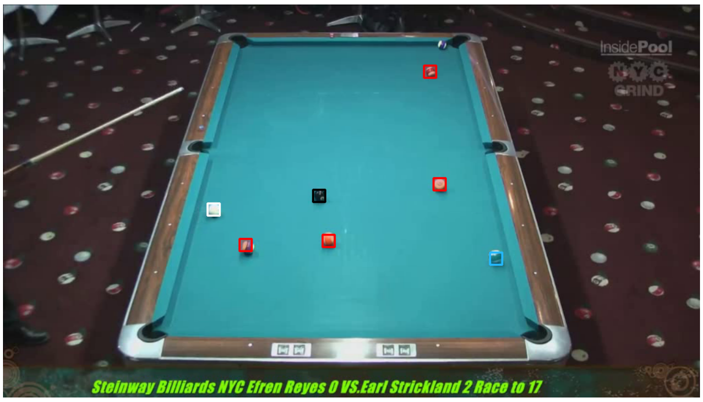


FIGURE 55: Object detection


FIGURE 56: Playing field border detection and segmentation

FIGURE 57: Game 4, clip 1, last frame: $mAP = 63.636364\%$, $mIoU = 62.259338\%$

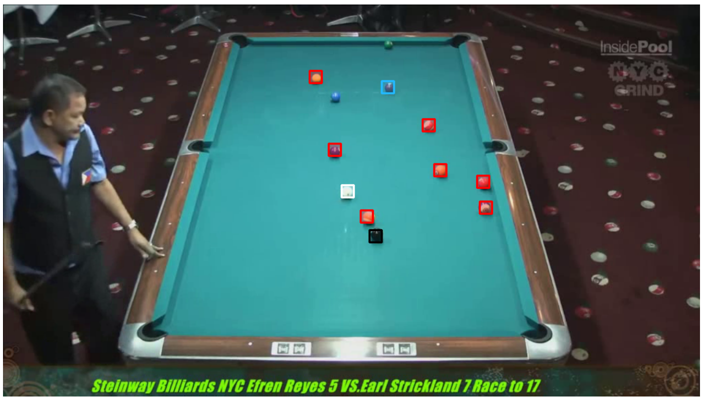


Figure 58: Object detection


Figure 59: Playing field border detection and segmentation

Figure 60: Game 4, clip 2, last frame: $mAP = 79.545455\%$, $mIoU = 60.517731\%$

### 8.3 2D top-view visualization map of last frame of each video

FIGURE 61: Game 1, clip 1, last frame: 2D top-view map

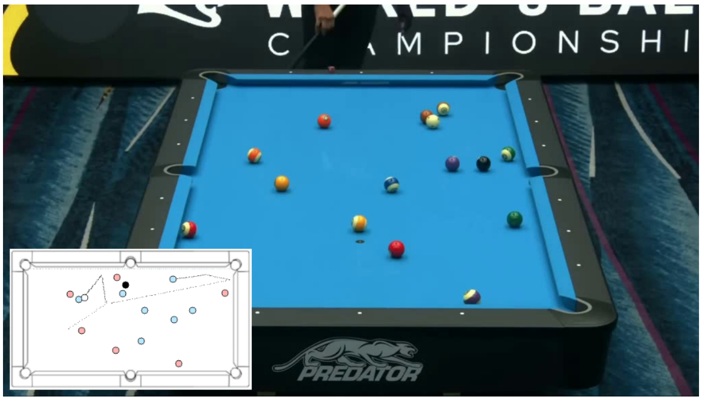


FIGURE 62: Game 1, clip 2, last frame: 2D top-view map

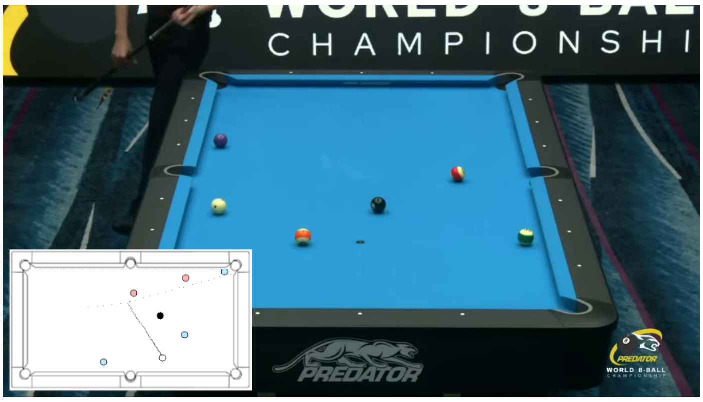


FIGURE 63: Game 1, clip 3, last frame: 2D top-view map


FIGURE 64: Game 1, clip 4, last frame: 2D top-view map

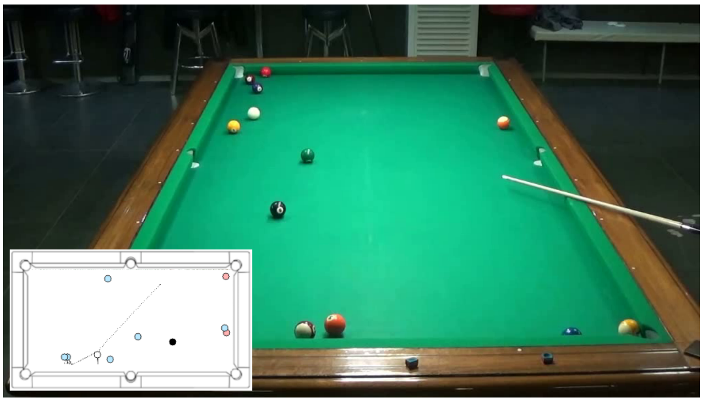

FIGURE 65: Game 2, clip 1, last frame: 2D top-view map

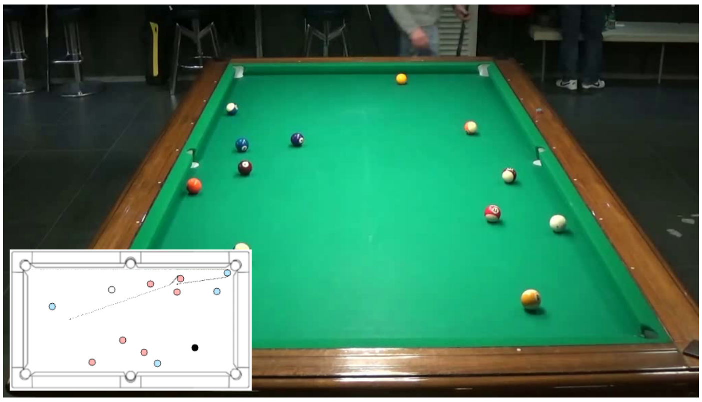

FIGURE 66: Game 2, clip 2, last frame: 2D top-view map

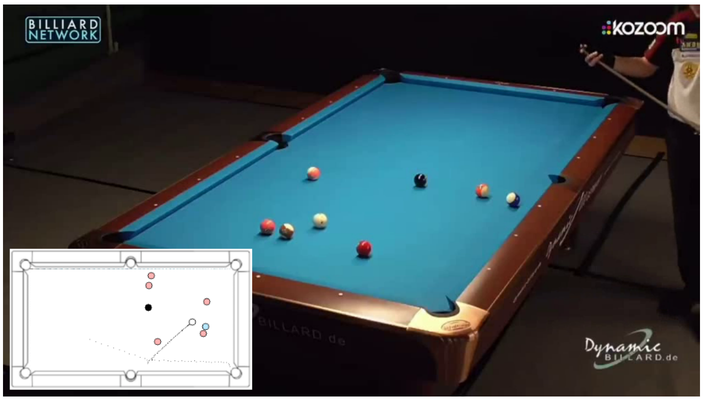


FIGURE 67: Game 3, clip 1, last frame: 2D top-view map

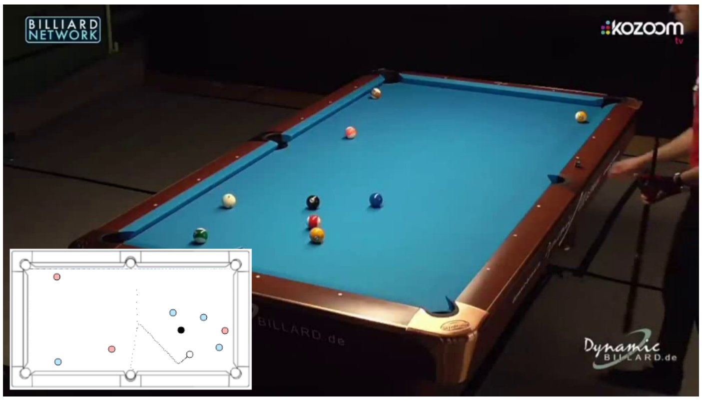


FIGURE 68: Game 3, clip 2, last frame: 2D top-view map

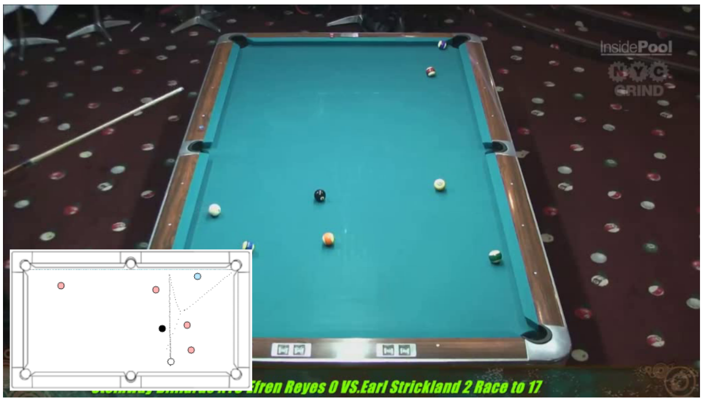


FIGURE 69: Game 4, clip 1, last frame: 2D top-view map


FIGURE 70: Game 4, clip 2, last frame: 2D top-view map