



# 3D reconstruction system and multiobject local tracking algorithm designed for billiards

Francisco J. Rodriguez-Lozano<sup>1</sup> · Juan C. Gámez-Granados<sup>1</sup> · Héctor Martínez<sup>1</sup> · Jose M. Palomares<sup>1</sup> · Joaquín Olivares<sup>1</sup>

Accepted: 23 February 2023 / Published online: 6 June 2023  
© The Author(s) 2023

## Abstract

The use of virtual reality or augmented reality systems in billiards sports are useful tools for pure entertainment or improving the player's skills. Depending on the purpose of these systems, tracking algorithms based on computer vision must be used. These algorithms are especially useful in systems aiming to reconstruct the trajectories followed by the balls after a strike. However, depending on the billiard modality, the problem of tracking multiple small identical objects, such as balls, is a complex task. In addition, when an amateur or nontop professional player uses low-frame-rate and low-resolution devices, problems such as blurred balls, blurred contours, or fuzzy edges, among others, arise. These effects have a negative impact on ball-tracking accuracy and reconstruction quality. Thus, this work proposes two contributions. The first contribution is a new tracking algorithm called “*multiobject local tracking (MOLT)*”. This algorithm can track balls with high precision and accuracy even with motion blur caused by low-resolution and low-frame-rate devices. Moreover, the proposed MOLT algorithm is compared with nine tracking methods and four different metrics, outperforming the rest of the methods in the majority of the cases and providing a robust solution. The second contribution is a whole system to track (using the MOLT algorithm) and reconstruct the movements of the balls on a billiard table in a 3D virtual world using computer vision. The proposed system covers all steps from image capture to 3D reconstruction. The 3D reconstruction results have been qualitatively evaluated by different users through a series of questionnaires, obtaining an overall score of 7.6 (out of 10), which indicates that the system is a promising and useful tool for training. Finally, both the MOLT algorithm and the reconstruction system are tested in three billiard modalities: blackball, carom billiards, and snooker.

**Keywords** Multi-object Tracking · Computer vision · Virtual reality · Billiard · 3D Reconstruction

## 1 Introduction

Object tracking algorithms [1–3] have a very wide field of applicability frequently being utilized in games and sports [4–8]. Although these algorithms usually work reasonably well, there are cases that make their task complex. This especially happens in pool games, also known as billiard games. Both terms refer to games played on a table with a cue (white ball). However, the difference is that the first term was originally proposed for tables with pockets and the second without pockets, although currently the term

billiards can be considered the most widespread to refer to all types of games [9].

Nevertheless, billiards is not only a well-known game played throughout the world but also a sport due to the high degree of practice that is required in addition to the effects that the physical effort exerts on the musculature of the players' bodies [10–12]. Such popularity is reached by this sport which is found not only in bars and recreational centers but also in professional players who compete in world tournaments. In fact, unlike other sports, billiards is a sport with a multitude of modalities [11, 13]. Examples of modalities are carom billiards [14], snooker [15], and blackball [16]. For this reason, given the importance of this sport, there have also been proposals to include billiards in the Olympic Games in Paris in 2024 [17].

Many billiard practitioners capture games in a video to analyze the movements and techniques used in each shot. These players require accuracy systems that provide precise

✉ Francisco J. Rodriguez-Lozano  
fj.rodriguez@uco.es

ball positioning and tracking to accurately reproduce the movements of each ball. With a proper 3D reconstruction of each shot, the players can change the point of view from the top view, which is the most usual capture angle from cameras placed on top of the table, to any angle. Thus, the player can recognize which shot is the most recommended and how to perform that shot. The 3D reconstruction of the shots requires tracking the trajectories of every ball on the table. It is very common that many balls move in one shot. These balls are very similar; in some modalities, the color varies among balls, but in other modalities, there is no change, even in the color. The similarity of the balls implies that tracking these objects, without any specific distinguishing elements, is a very demanding and complex task. Moreover, the typical devices used by amateur and nontop professional players cannot catch all the details in a shot due to low-frame-rate and low-resolution limitations.

The aforementioned limitations mean that the tracking algorithms have to work under extremely unfavorable conditions. Some of the conditions are ball motion blurring due to their velocity, fuzzy ball edges, or very abrupt displacements between two consecutive frames. All these unfavorable conditions are further magnified when the number of balls in play is high. Specifically, snooker [15] and blackball [16] are examples with multiple small identical objects of the same color that increase the complexity of tracking object movements. Therefore, accurately tracking the different objects for reconstruction under these conditions is a task where many existing tracking algorithms cannot work properly.

In this context, this work proposes two contributions. The first contribution is a new multiobject tracking algorithm with local trackers (MOLT), which solves the tracking of multiple identical targets (both shapes and colors) and can work with devices with low computational capacity and reduced image quality and refresh rate. The second contribution is a complete system to reconstruct the billiard shots in a 3D-generated virtual world that can be used in training systems or for entertainment purposes. Consequently, by combining both proposals, this work provides a system that covers the necessary steps to be applied from image capture, billiard table detection, ball tracking, and the 3D reconstruction of three widely played modalities of billiards: blackball, carom billiards, and snooker.

This paper is organized as follows. Section 2 summarizes those proposals existing in the scientific literature that are focused on one or several parts of the aim of this work related to any modality of billiard games. Section 3 describes the proposed system, which includes preprocessing methods, baize segmentation, ball detection and classification, the MOLT algorithm, and the 3D recreation of the shots. The results and comparisons with other methods

are analyzed and discussed in Section 4. Finally, the main conclusions and future works are presented in Section 5.

## 2 Related works

The problem of information extraction from billiard games has been addressed by different authors in the scientific literature with the aim of developing algorithms for object segmentation, object tracking, and systems focused on training.

In this context, Ling et al. [18] performed multiple object detection in snooker games. The authors carried out snooker table identification by color segmentation of images from a video recording. Ball detection was performed in two stages, as green balls showed detection problems due to the similarity with the background color, as noted by the authors. Nongreen balls were found using the watershed algorithm [19] and color segmentation. Green balls were detected by analyzing the illumination reflections.

A similar idea of locating the ball position through the illumination reflection was used by Legg et al. [20]. In their work, the authors performed a table detection technique by transforming the images obtained by the camera into HSL (Hue, Saturation, and Lightness) color space and obtaining a binary mask to detect the playing area. Ball tracking was carried out frame by frame, relying on the reflection of light on the balls and on the minimum distance of a ball detected in the previous frame to estimate its current position. These procedures of table detection, identification, and ball tracking were later used by Parry et al. [21] for a study of hierarchical event selection, generating storyboards representing moments of change or key events.

Vachaspati [22] proposed a system to identify the billiard balls positioned on the baize. For this purpose, the author used a billiard table detection technique by extracting the predominant color [23–26] of the image in the HSL color space. This color space further allows identification and estimation of the position of each ball based on its color and thus tracking the balls frame by frame.

Other solutions, such as those proposed by Baekdahl and Have [23], Weatherford [24] and Hsu et al. [27], employed the same idea of baize segmentation using the predominant color of the image and proposed a system to identify solid and striped balls used in other billiard game modalities. This ball identification is achieved by subtracting the background using the color of the baize area.

Other authors focused on detecting and tracking a special type of ball, striped, red, and yellow balls, or only identifying and tracking the cue ball. For example, in the proposal of Larsen et al. [28], only yellow balls and the cue ball are identified. In fact, other proposals, such as the

work of Sousa et al. [29], detect and track only the cue ball to analyze its collisions and interactions with other balls, using a background subtraction approach to determine its position. Another cue ball identification proposal was presented by Gao et al. [30], where the position of the ball after a strike and collision is predicted using a neural network (NN) method in conjunction with a fuzzy dynamic model.

Park and Park [31] proposed using the CAMshift algorithm [32] for cue ball tracking (the other balls are not considered), for which the authors carried out the billiard table detection using the Harris method. A similar proposal in which patterns on the billiard table are considered is found in the work of Larsen et al. [28], where the baize is identified using patterns that are usually present on many billiard tables, such as diamonds on the rails (wooden edge) of the table.

Gao et al. [33] proposed a system for the recognition of different balls in 8-ball billiards. Their system incorporates the use of a CCD camera positioned above the table, and through the use of computer vision algorithms and artificial intelligence, they are able to identify the objects. The authors perform a segmentation of the baize based on frequency filtering in the RGB color space, taking into consideration that the baize is the most common color (green). For the localization of the balls, they use an improved version of the Hough transform [34] together with the least squares (LS) method. Once each ball position is detected, the classification is performed by means of a convolutional neural network (CNN) [35].

In the literature, we can find studies that are focused on extracting information using physical models [12, 36]. An example is found in the work of Gabdulkhakova and Kropatsch [37], where the authors proposed a model for analyzing games played on a snooker table. The analysis is based on generating a kinematic model to predict billiard ball motion using physical features related to ball movements.

Another work extracting parameters of billiard ball dynamics and physical models is found in the proposals of Mathavan et al. [38], where these models were used in later works [39] by the authors to develop a robotic system that mimics the behavior of professional billiard players. Robotics systems have also been explored by Tung et al [40], who used recorded videos of professional billiard players to train a machine learning algorithm as the robot brain. Another work focused on robotics systems was proposed by Bhagat [41] to find the optimal shot trajectory to pot a ball of a certain color verbally identified by the player into a pocket using deep learning algorithms.

Other recent approaches proposed using control algorithms based on closed-loop systems [42–44], in which the billiard ball trajectories were mapped to an infinite surface wherein impacts never occur, and then a position feedback

controller was designed for trajectories evolving on this surface. However, these approaches are not based on computer vision but on predicting trajectories in a simulated environment.

In addition, in the billiards context, we also found some interesting proposals aimed at extracting information [45, 46] and helping beginners understand this sport [47]. Therefore, the works of Legg et al. [20] and Parry et al. [21] mentioned above were taken as the basis for developing of a test-based skill training system proposed by Chung et al. [48]. This type of extracted information is not only useful for teaching new players but can also be used to evaluate the different rating systems for professional players, as seen in the work of Collingwood et al. [49]. Another example of the use of extracted information together with artificial intelligence algorithms was presented in the work of Li et al. [46] where the authors focused on using artificial neural networks (ANNs) to predict the outcome of professional snooker games.

Moreover, as an aid system for beginner players, Sun et al. [47] proposed the GraspSnooker system, a tool, that by using shot strategy predictors and text generators for automatically creating snooker game comments, helps users to understand the different events that occur in the games.

To achieve better billiard player training, techniques such as shot prediction and augmented reality can be used. In this context, Jebara et al. [50] presented a training system to determine shot direction based on predicting ball interactions. Their study, employed wearable devices to endow the system with augmented reality. A modern study with the same objective but with more current, devices such as Microsoft HoloLens can be seen in Medved [51].

Shih [52] proposed a low-cost game training system for billiards and shot prediction. Table detection was performed using a chessboard calibration pattern. Ball detection was performed by a background subtraction algorithm using the baize color. This author conducted another study [53] in which three different planning strategies were compared to analyze the effect of the shot using an augmented reality system.

Sousa et al. [29] developed another augmented reality system to train inexperienced billiard players. The system consists of making a shot prediction and the movement of the balls to be struck so that the players have information about what is going to happen before making the shot. To do this, like many of the previous proposals, they perform a method to subtract the baize of the image, detect the balls and identify only the cue ball.

Paolis et al. [54] proposed a virtual reality application as a prototype of a billiard training system. This application generates a 3D virtual world space where the cue and cue ball are reconstructed using markers placed on the tip of the cue and on a flat surface.

However, training systems are not only focused on the shot direction or on the recreation of ball movements, but we find proposals such as the work of Mishima and Suganuma [55], where a support system for beginners starting to play billiards was developed. The aim of the system was to provide information to improve the player's shooting stance using RGB-D sensors (Red-Green-Blue-Depth).

Another example of this use of sensors applied to a billiard training system can be found in the proposal by Pinzon et al. [56], where they developed a system that predicts the shot direction in augmented reality billiard applications. Unlike other proposals that use this type of sensor, these authors obtain the region of the baize by applying the Hough transform to obtain the lines and corners of the table based on the image generated by the depth sensor. The balls are also obtained by making exclusive use of this sensor using the relative ball height with the baize. In addition, the billiard cue detection is used to estimate the possible trajectory that the balls will follow once they are struck.

Finally, 3D computing vision techniques are also used to help players in their training. Therefore, Wu and Dellinger [57] proposed a mixed-reality system to simulate the billiard game in a 3D world displayed on a large screen placed next to the billiard table. Their work uses an RGB-D sensor to detect the player and the cue. Additionally, their work uses depth information together with color information to detect the ball position through the Hough transform method. Kato et al. [58] proposed the *OpenPool* framework, an open platform composed of three different libraries where the main aim is the generation of visual effects of the collisions between the billiard balls to improve the game with the use of augmented reality. This framework also uses a depth sensor to segment the table and identifies when a ball is pocketed through infrared emitters located in each pocket using IoT communications protocols such as ZigBee [59].

Once the proposals of the authors have been reviewed, the following list highlights the contributions of this work to design a system for entertainment and player training:

- A system capable of being used in different billiard modalities, although not limited to blackball, carom billiards, and snooker.
- A highly modular system with the following elements:
  - A module for baize segmentation that is invariant to its color and does not require table-specific features such as diamonds, pockets, or manufactured identifiers such as chessboard patterns.
  - A module for ball identification and classification invariant to ball size.

- A new multiobject tracking module to obtain the ball movements and positions at low-resolution and low-frame-rate videos. Additionally, the tracking algorithm for balls is designed to be robust to occlusions and ball blurring due to rapid ball movement.
- A module to generate a 3D world to reconstruct the billiard shots.

In addition, Table 1 summarizes the main differences between the detailed works of the previous authors and this proposed work.

### 3 Proposed MOLT algorithm and system

The main objectives of this work are twofold. The first objective is a new tracking algorithm called multiobject local tracker (MOLT) that is proposed and described in Section 3.2. Tracking every ball is a difficult task, especially in environments with low frame rates and low-quality image capture sensors. Therefore, to carry out this tracking, several preprocessing steps are necessary, which are detailed in Section 3.1. The second objective is a whole system to reconstruct billiard shots, collisions and ball movements in a 3D virtual world (Section 3.3). This objective is performed using several preprocessing steps, the MOLT output and postprocessing steps.

#### 3.1 Preprocessing methods

In this subsection, several steps are included to preprocess the raw input images. These images may suffer from poor quality due to the usage of low-resolution cameras. These devices may produce objects in which the contours are not accurate or even completely fuzzy. Therefore, all these actions described in this section are used to provide the best initial inputs to the MOLT algorithm.

##### 3.1.1 Previous information

The proposal presented in this paper covers three billiard modalities: blackball, carom billiards, and snooker. However, the system permits different game modalities because the number, color, and size of the balls is not a restricting factor. To achieve this versatility, it is necessary to provide the system with the following information:

- Ball colors.
- Ball diameters.
- Baize dimensions.
- Pocket sizes if they exist.

**Table 1** Summary of comparison of objectives with other authors' proposals

Work	Shot prediction	Ball tracking	Ball identification	Baize color/patterns segmentation	Augmented 3D reality system	3D Reconstruction	Information extraction	Billiard modalities
Ling et al. [18]	N	N	Y	Y	N	N	N	Snooker
Legg et al. [20]	N	Y	Y	Y	N	Y	N	Snooker
Parry et al. [21]	N	Y*	Y*	Y*	N	Y*	Y	Snooker
Chung et al. [48]	N	N	N	N	N	N	Y	Snooker
Vachaspati [22]	N	Y	Y	Y	N	N	N	9-ball
Baekdahl and Have [23]	Y	N	Y	Y	N	N	N	8-ball
Weatherford [24]	Y	N	Y	Y	N	N	N	8-ball
Uchiyama and Saito [25]	Y	N	Y	Y	Y	N	N	9-ball
Hsu et al. [27]	N	N	Y	Y	Y	N	N	Snooker
Larsen et al. [28]	Y	N	Y	Y	Y	N	N	-
Sousa et al. [29]	Y	N	N	N	Y	N	N	8-ball
Park and Park [31]	N	Y**	Y**	Y	N	N	N	Carom billiards
Gao et al. [33]	N	N	Y	N	N	N	N	8-ball
Jebara et al. [50]	Y	N	Y	Y	Y	N	N	8-ball
Shih [52]	Y	N	Y**	Y	N	N	N	-
Shih [53]	Y	N	Y**	Y	Y	N	N	-
Paolis et al. [54]	N	N	N	N	Y	Y	N	Carom billiards
Mishima and Sukanuma [55]	N	N	N	N	N	N	Y	-
Pinzon et al. [56]	Y	N	N	N	Y	N	N	Carom billiards
Kato et al. [58]	-	Y	-	Y	Y	N	N	-
Gabdulkhakova and Kropatsch [37]	Y	Y	-	-	N	N	N	Snooker
Mathavan et al. [38]	Y	N	N	Y	N	N	Y	Snooker
Mathavan et al. [39]	Y	N	N	N	N	N	Y	Snooker
Yu et al. [45]	N	N	N	Y	N	N	Y	Snooker
Li et al. [46]	N	N	N	N	N	N	Y	Snooker
Sun et al. [47]	N	-	-	-	N	N	Y	Snooker
Wu and Dellinger [57]	Y	Y**	Y	-	N	Y	N	-
Menini et al. [42]	-	Y	-	-	N	N	Y	-
Menini et al. [43]	-	Y	-	-	N	N	Y	-
Menini et al. [44]	-	Y	-	-	N	N	Y	-
Tung et al. [40]	Y	N	Y**	N	N	N	Y	8-ball
Supriadi et al. [12]	N	N	Y**	N	N	N	Y	-
Cross [36]	N	N	N	N	N	N	Y	-
Gao et al. [30]	Y	N	Y**	N	N	N	Y	Snooker
Medved [51]	Y	N	Y**	Y	Y	N	N	9-ball
Collingwood et al. [49]	-	-	-	-	-	-	Y	Snooker
Bhagat [41]	Y	N	Y	Y	-	N	N	Snooker
Rodriguez-Lozano et al. (This work)	N	Y	Y	N	N	Y	N	Blackball Carom billiards Snooker

Note: Yes and No are specified with {Y, N}, and when not enough information is provided by authors it is specified with {-}.

\*Based on Legg et al. [20] work.

\*\* Only the cue ball is identified or tracked.

- Approximate height of the video camera located above the baize.

This information is generically preset for each game modality (blackball, carom billiards, and snooker) with the standard table and ball measurements. Hence, this information should be changed, for example, if the balls are of a nonstandard size or the table dimensions are different. During the image acquisition step, the proposed system considers that the images have been calibrated to correct possible distortions caused by the camera lenses. This can easily be achieved using methods provided by computer vision libraries such as OpenCV [60] or similar.

### 3.1.2 Baize segmentation

The first step is to delimit the playable area of the table. However, the whole table is not a playable area, since the balls can only move within the baize area. It is, therefore, necessary to delimit that area of the image and discard the unnecessary information. For detecting the baize, many authors, as described in Section 2 and summarized in Table 1, proposed mechanisms based on baize color or even on pattern detection, such as the table diamonds or pocket locations. However, since this work is designed for different billiard modalities, a method that accounts for the use of different table types is necessary. For this reason, the method cannot be based on predominant baize color or on patterns such as pockets or diamonds because they may not exist. Therefore, this work proposes the mechanism shown in Procedure 1 to solve this step:

- 
- 1: **Inputs:** (*frame*, *baize\_size*)
  - 2: *edge\_frame*  $\leftarrow$  Edge\_Detection (*frame*)
  - 3: *lines*  $\leftarrow$  Lines\_Detection (*edge\_frame*)
  - 4: *intersec\_points*  $\leftarrow$  Compute\_Intersections (*lines*)
  - 5: *vertices\_points*  $\leftarrow$  Find\_Vertices (*intersec\_points*)
  - 6: *baize\_area*  $\leftarrow$  Get\_baize\_Area (*vertices\_points*, *baize\_size*)
  - 7: *baize\_image*  $\leftarrow$  Generate\_ROI (*frame*, *baize\_area*)
  - 8: **Output** *baize\_image*
- 

**Procedure 1** Steps to obtain the baize area.

Procedure 1 has two inputs: the “*frame*”, which corresponds to an image taken by the camera, and “*baize\_size*”, which is the dimension of the baize. These variables are used within the procedure as follows:

1. Edge\_Detection: The Canny method [61] is applied to obtain all the edges of the input image. After edge detection, a binary dilation method is applied to obtain

more robust results in the next step. The result is a new image that contains only all the detected edges.

2. Lines\_Detection: The Hough transform [62] is applied to detect all the possible lines and their equations from the edge image.
3. Compute\_Intersections: The intersection points are computed using the calculated line equations. This step generates a list of intersection points between all the lines. Additionally, all the intersections that lie outside the image size are discarded.
4. Find\_Vertices: All the points of the previous step are checked to select groups of four points that create a rectangle among them.
5. Get\_baize\_Area: To delimit the playable area, the next step is to discard all found rectangles not corresponding to the baize area using the “*baize\_size*”.
6. Generate\_ROI: The final step is to obtain a region of interest from the original image containing only the playable area (the baize).

Finally, a graphical representation of the results of the proposed Procedure 1 can be seen on the left of Fig. 1.

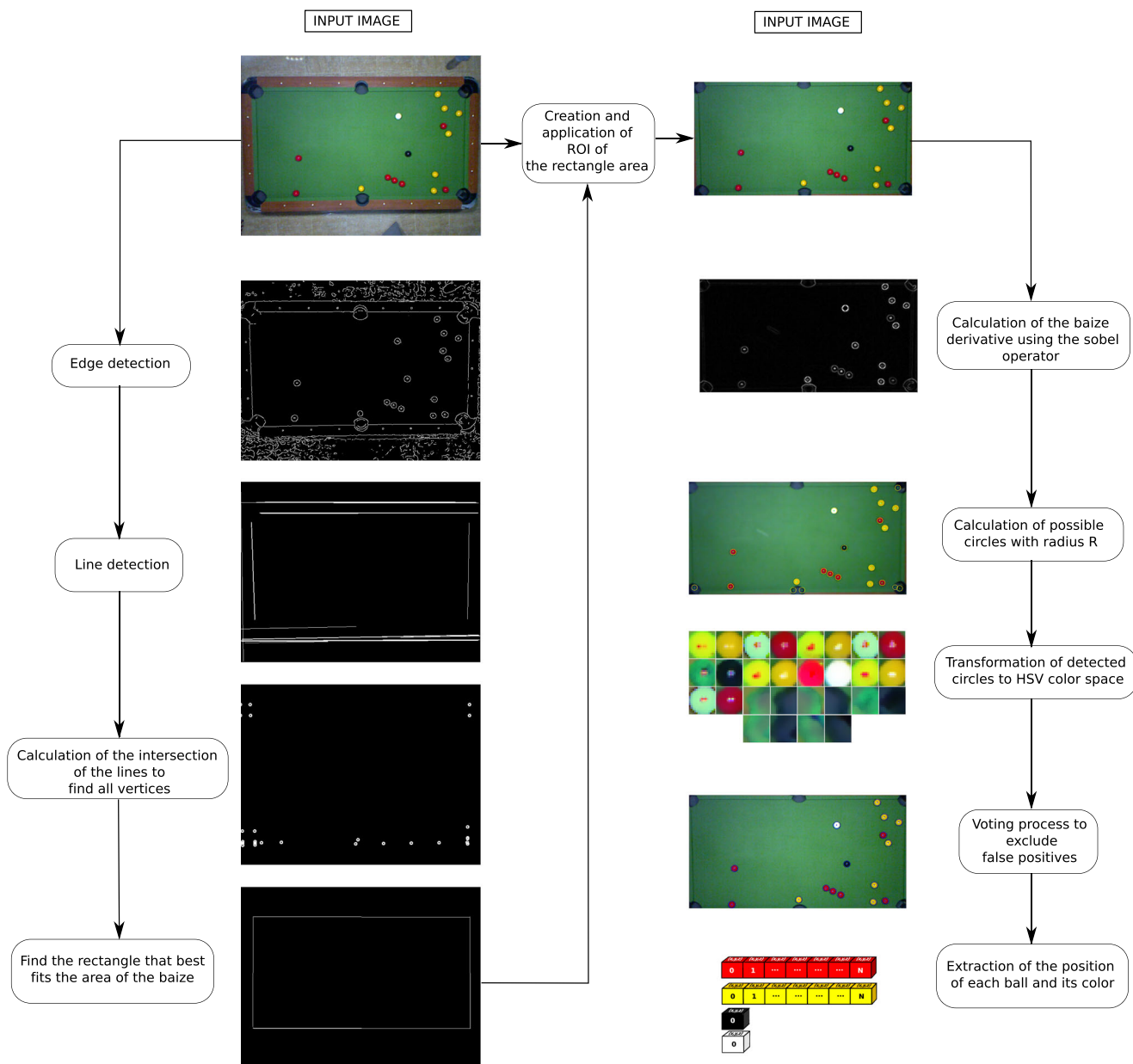
### 3.1.3 Ball detection and identification

The next step of the proposed system is to obtain the location of the different balls found on the baize. Additionally, to allow for different types of billiard modalities, the number of balls is not restricted. Therefore, this stage determines the position and color of the balls. To accomplish this, Procedure 2 is proposed, where the region of interest of the baize (*frame\_roi*), the color of the balls (*ball\_colors*) and their size (*ball\_size*) are used as inputs:

- 
- 1: **Inputs:** (*frame\_roi*, *ball\_colors*, *ball\_size*)
  - 2: *derivFrame*  $\leftarrow$  Derivatives (*frame\_roi*)
  - 3: *circle\_positions*  $\leftarrow$  Circle\_Detection (*derivFrame*, *ball\_size*)
  - 4: *ball\_list*  $\leftarrow$  HSV\_Transformation (*frame\_roi*, *circle\_positions*)
  - 5: *ball\_positions*  $\leftarrow$  Discard\_False\_Balls (*ball\_list*, *ball\_colors*)
  - 6: **Output** *ball\_positions*
- 

**Procedure 2** Steps to obtain initial ball positions.

1. Derivatives: The baize region of interest obtained in Section 3.1.2 contains the balls. However, to facilitate calculating their position, the Sobel method [63] is used to simplify the image by obtaining a new image with the gradient of the derivatives (*derivFrame*).



**Fig. 1** Baize segmentation and ball identification steps

2. **Circle\_Detection:** The next step is to apply the Hough method [62, 64], but in this case, to obtain the possible circles of the image. To do this, we calculate the radius (in pixels) of the balls, which is obtained from the size of the balls (*ball\_size*). Finally, a list of the positions of each detected circle is obtained.
3. **HSV\_Transformation:** To simplify the color identification process of each ball, we transform the image from the RGB color space to the HSV color space [65], which allows us to split the color values from the illumination values more easily than with the RGB model. This step extends the previous list by adding to the position of the circles their HSV image (*ball\_list*).
4. **Discard\_False\_Balls:** In the previous steps, the position of each detected circle was obtained, but it may be the case that other circles are detected as balls (as in the case of pockets if they exist). Therefore, this last step has the objective of discarding all those circles that do not in reality correspond to a ball. To do this, and since the color of each ball in play is known (*ball\_colors*), a voting process is performed to determine how much color each of the detected circles has in comparison to the defined colors. In this case, as the game modalities in which the system will be tested are blackball, carom billiards, and snooker, the possible colors are different, and a color range must be defined to discard false

positives. This work uses the HSV color space to analyze the color of each ball candidate. Note that the color range values for the HSV space used in this work are  $[0, 180]$  for the  $H$  channel and  $[0, 255]$  for the  $S$  and  $V$  channels:

- Blackball: For blackball, there are yellow, red, black and white ball colors:
  - $H[120 - 179]$ ,  $S[180 - 255]$ ,  $V[100 - 200]$  for red balls.
  - $H[15 - 30]$ ,  $V[160 - 255]$  for yellow balls.
  - $V[0 - 50]$  for the black ball.
  - $S[0 - 50]$  for the cue ball.
- Snooker: In this modality, the colors, in addition to blackball, are pink, brown, blue, and green:
  - $H[130 - 179]$ ,  $V[190 - 255]$  for the pink ball.
  - $H[0 - 50]$ ,  $S[0 - 150]$ ,  $V[0 - 180]$  for the brown ball.
  - $H[90 - 140]$ ,  $S[150 - 255]$ ,  $V[150 - 200]$  for the blue ball.
  - $H[80 - 100]$ ,  $S[150 - 255]$  for the green ball.
- Carom billiards: In this modality, the colors are the same as those used for blackball with no black.

Once the number of matching pixels within each circle has been checked, it is possible to eliminate false positives. An example of the blackball modality can be seen on the right of Fig. 1. Note that this proposed step permits the use of different color balls, and if the user wants to use nonstandard colors, these can be defined here.

At the end of Procedure 2, a data structure of the balls differentiated by color is obtained. This structure is represented graphically in the bottom right of Fig. 1. Only the blackball example is shown to limit the number of elements in the figure. Additionally, these balls are the different objects to be tracked by the MOLT algorithm proposed in the next section.

### 3.2 Multiobject local trackers algorithm (MOLT)

The new proposed multiobject tracking algorithm uses local trackers (MOLT). This algorithm works deterministically to calculate the position of the objects being tracked. For example, if we have an object at time instant  $t_n$ , the algorithm will determine the position of the tracked object based on the information known at time  $t_{n-1}$ . To achieve this, the MOLT algorithm assigns to each tracked object a population of trackers. The trackers can be considered small regions or windows that search for similar information in a

delimited environment (*exploration radius*). Each tracker is a structure composed of the following elements:

- *Center point*: Corresponds to the central position of the tracker. In the case of the blackball and snooker modalities, since balls can fall into a pocket, each tracker has a component on each axis ( $x, y, z$ ).
- *Size*: Determines the size of the tracker from the center point. This value establishes the size of the tracked object acting as a radius from the *center point*.
- *Histogram weight* corresponds to the degree of similarity between the histogram of the object before tracking and the histogram of the tracker. This degree of similarity is in the range  $[0-1]$ , where 0 is no similarity and 1 is the maximum similarity.
- *Distance weight* determines the distance ratio between the object tracked at a previous time instant and the tracker. This variable is also represented in the range  $[0-1]$ , where 0 represents a far distance between objects, and 1 represents a close distance.
- *Total weight*: Weighted sum of the histogram weight and distance weight variables. This variable is also in the range  $[0-1]$  provided by  $total\ weight = \alpha * histogram\ weight + (1 - \alpha) * distance\ weight$ .

Each population of trackers generates a certain number of local trackers within the *exploration radius* that specifies the maximum scanning range of the trackers. These values can be the same for all tracked objects or different for each of them. In the top right of Fig. 2, an example of the assignation of tracker populations by balls can be seen in the data structure.

Therefore, as shown in Procedure 3, the MOLT algorithm receives three parameters: a list of consecutive images to track the objects (*frame\_list*), the position in the initial state of the balls (*balls\_positions*), and the size of each ball (*balls\_size*). Note that the steps above the MOLT tracking algorithm (*Procedures 1 and 2*) are performed only once for the first frame captured by the camera, and these methods are not employed again until another shot is taken.

The following subsections detail the methods used in Procedure 3.

#### 3.2.1 Initialization step (Init\_Structure)

The first step of the MOLT algorithm consists of initializing the data structures of the objects to be tracked. In this step, a population of trackers is assigned to each tracked object, and this is only performed the first time the algorithm is executed. The specific tasks to be performed in this step are detailed in Procedure 4, which receives the first *frame* and the ball features (*balls\_positions*, *balls\_size*) as inputs:



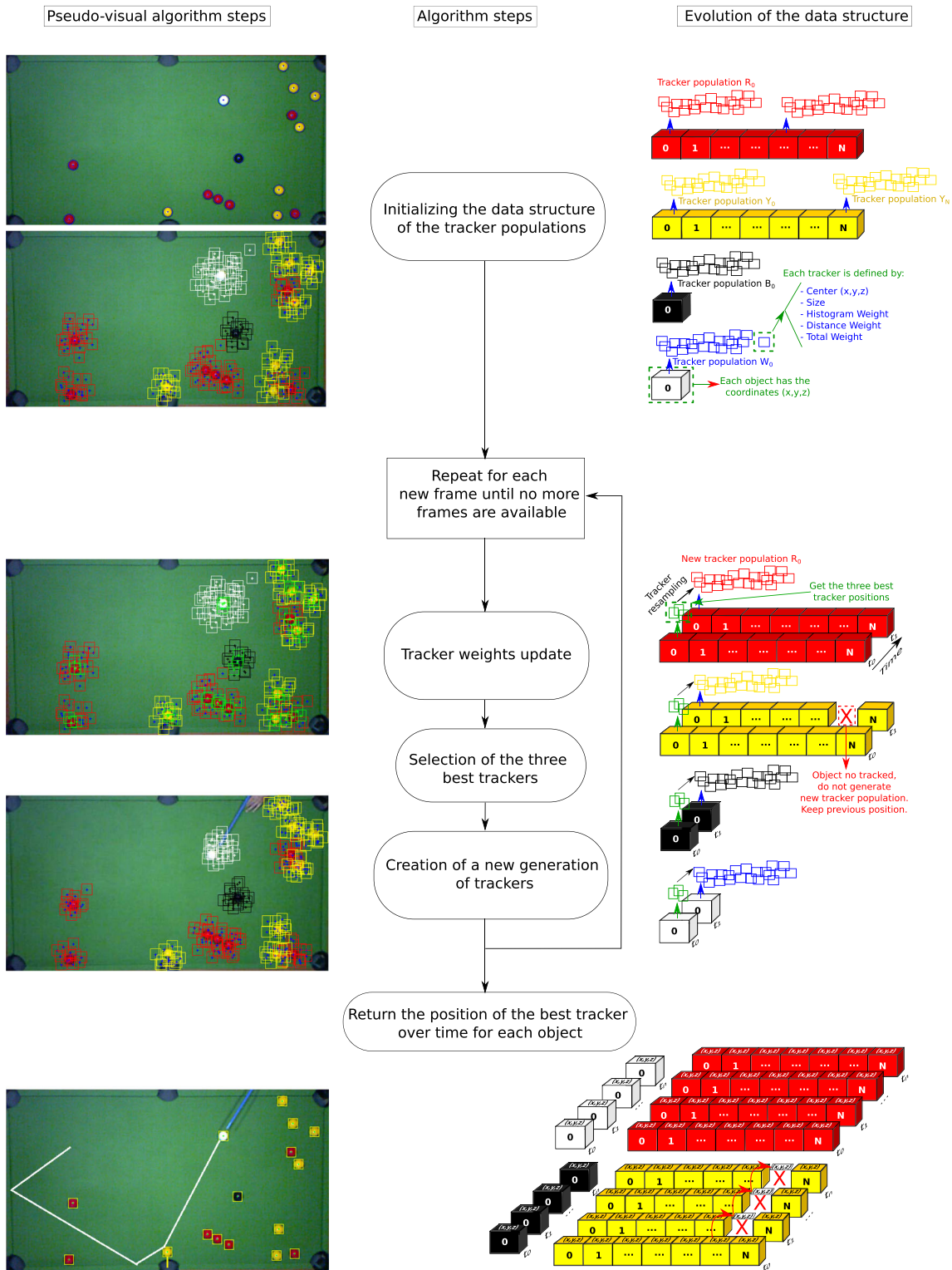


Fig. 2 MOLT algorithm visual steps

- The histogram of each detected ball is calculated and stored in *ball\_histograms* to be used in the next procedure. Additionally, a number *n* of trackers is assigned. This parameter can be defined by the user and can be different for each object to be tracked.
- For each tracker within the population of trackers,

---

```

1: Inputs: (frame_list, balls_positions, balls_size)
2: firsttime = true
3: while frame_list  $\neq$  empty do
4:   frame  $\leftarrow$  pop_front(frame_list)
5:   if firsttime == true then
6:     tracker_population, ball_histograms  $\leftarrow$  Init_Structure (frame, balls_positions, balls_size)
7:     firsttime = false
8:   end if
9:   Update_Population (frame, tracker_population, ball_histograms)
10:  balls_tracking  $\leftarrow$  Get_Best_Tracker (tracker_population)
11:  Resample_Population (tracker_population, exploration_radius)
12: end while
13: Output balls_tracking

```

---

**Procedure 3** Main steps of Multi-Object local trackers algorithm (MOLT).

---

```

1: Inputs: (frame, balls_positions, balls_size)
2: for each ball in balls_positions do
3:   ball_histograms  $\leftarrow$  compute_histogram(frame,
4:   balls_positions)
5:   tracker_population  $\leftarrow$  generate_trackers_population(n)
6:   for  $t \leftarrow 1, 2, 3, \dots, n$  do
7:     tracker_population[t].Center_point  $\leftarrow$  [ball.x,
8:     ball.y, 0]
9:     tracker_population[t].Size  $\leftarrow$  balls_size
10:    tracker_population[t].Histogram_weight  $\leftarrow$  1
11:    tracker_population[t].distance_weight  $\leftarrow$  1
12:    tracker_population[t].total_weight  $\leftarrow$  1
13:   end for
14: end for
15: Output tracker_population, ball_histograms

```

---

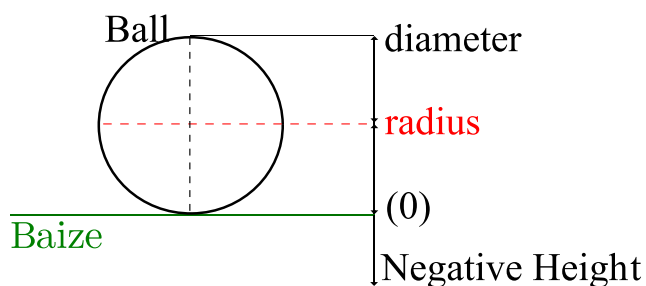
**Procedure 4** Structure initialization step.

assign the initial  $x$  and  $y$  positions of the ball. The  $z$  coordinate is initialized to 0 since the balls above the baize are at a height of 0, as shown in Fig. 3.

- Each tracker is initialized with the size of the object to be tracked (the balls) and the best possible weight.

### 3.2.2 Update population information (Update\_Population)

This step of the algorithm is one of the most important since it is where the tracker information is updated. To perform this update, the current frame, the tracker population and



**Fig. 3** Ball heights

the histogram of the tracked objects are used as inputs in Procedure 5 as follows:

First, from each population, the best tracker from the previous frame of the algorithm is copied for comparison with the trackers of the current frame. For the first iteration of the algorithm after initialization (Procedure 4), any tracker in the population is the best. However, for subsequent iterations, the most accurate tracker is always the first because the trackers are ordered from most to least precise (Procedure 6).

After making the copy of the best tracker, this tracker is checked to determine whether the  $z$  coordinate of the tracker is greater than or equal to 0. This condition allows us to determine if the ball is on the baize and if pockets exist, over a pocket (in the process of being potted) or potted. Therefore, two cases are distinguished:

- If the ball is on the baize (Line 4), it is necessary to update the information of each tracker in the current frame. To do this, the value of each of the tracker weights must be updated:
  - For the histogram-based weight, the difference between histograms is calculated using the Bhattacharyya distance metric [66]. This step determines the histogram similarity between the histogram of the tracker image at the initial instant and the tracker at the current frame.
  - For the distance-based weight, the Euclidean distance between the  $(x, y)$  coordinates of the center points of the best tracker and the position of the current tracker is calculated. To perform comparisons, the value of this distance is in the range  $[0-1]$ . Therefore, its value is normalized by setting the maximum possible distance in two subsequent frames to be the diagonal of the baize.

---

```

1: Inputs: (frame, tracker_population, ball_histograms)
2: for each population of every ball do
3:   best_tracker  $\leftarrow$  tracker_population[1]
4:   if best_tracker.Center_point.z  $\geq$  0 then
5:     for t  $\leftarrow$  2,3,...n do
6:       tracker_population[t]  $\leftarrow$  Update_Histogram (frame, tracker_population[t], ball_histograms)
7:       tracker_population[t]  $\leftarrow$  Update_Distance (tracker_population[t], best_tracker)
8:       tracker_population[t].total_weight  $\leftarrow$  ( $\alpha$ *tracker_population[t].Histogram_weight)
          + ((1- $\alpha$ )*tracker_population[t].distance_weight)
9:     end for
10:  else[best_tracker.Center_point.z > -(best_tracker.size)]
11:    tracker_population[1].Center_point.z  $\leftarrow$  tracker_population[1].Center_point.z-1
12:  end if
13: end for

```

---

**Procedure 5** Update population information steps.

- The total weight determines the weighted average of the histogram and the distance. This accounts for the  $\alpha$  variable that determines the importance of the histogram weight. Thus, if the  $\alpha$  value is equal to 1, the distance weight is considered to be irrelevant for object tracking. The value of  $\alpha$  estimated for the ball-tracking problem is 0.5; in other words, both the histogram and the distance are of equal importance. This is because, in the case of identical color and shape, tracked objects (the balls) both provide relevant information about the motion between the position in the previous frame and the current frame.
- If the ball is not on the baize (Line 9), the  $z$  coordinate of the tracker is checked for negative values greater than the ball diameter. If this condition is satisfied, the  $z$  coordinate of the tracker is decreased by one unit. With this mechanism, the tracker that follows the ball once it enters a pocket decreases its  $z$  coordinate to a height below the baize (see Fig. 3). In addition, once the negative value of the condition is reached, for each potted ball, no new populations are generated. This effect can be seen graphically in the evolution of the data structure in Fig. 2. Note that this step is only performed for those game modalities that require a table with pockets.

### 3.2.3 Obtain best trackers (Get\_Best\_Tracker)

This step, as shown in Procedure 6, is responsible for sorting the trackers of each population (*tracker\_population*) using the *QuickSort* method [67]. In this way, the trackers are sorted from highest to lowest based on their total weight.

Additionally, this procedure, as seen in Lines 3 and 4, not only updates the order of the best trackers but also

---

```

1: Inputs: (tracker_population)
2: for each population of every ball do
3:   tracker_population  $\leftarrow$ 
     Quick_Sort (tracker_population)
4:   best_tracker_position_population
      $\leftarrow$  tracker_population[1].Center_point
5: end for
6: Output best_tracker_position_population

```

---

**Procedure 6** Steps to obtain the best trackers.

stores in each iteration the position of the best tracker along with the frames, thus obtaining the positions of the balls (*best\_tracker\_position\_population*). A representation of these positions can be seen at the bottom of Fig. 2.

### 3.2.4 Generate new populations (Resample\_Population)

The last step of the MOLT algorithm is the generation of new tracker positions. For this purpose, a “*diversity-oriented approach*” is proposed. In this approach, the position of the new trackers is not based solely on the position of the best tracker currently found but incorporates a percentage of the second-best tracker and the third-best tracker. This approach avoids the elitism of the algorithm and provides some diversity in the search. Specifically, as seen in Procedure 7, the new generation of trackers is based on the previous population (*tracker\_population*) received as an input of the procedure. Thus, 50% of the new positions of the trackers will be generated randomly within the limits (*exploration\_radius*) based on the location of the best tracker ( $bt_1$ ), 30% of the new positions based on the second-best tracker ( $bt_2$ ), and 20% of the new positions based on the third-best tracker ( $bt_3$ ). In addition, as in Procedure 5, a check is made to see if any of the balls have entered or

are over a pocket (Line 6) since in this case the object is no longer tracked and it is not necessary to generate trackers.

Once this procedure is finished, the new generation of trackers is ready to be analyzed in the next frame, and this procedure is repeated as long as there are future frames, as shown in the loop in Procedure 3.

### 3.3 3D reconstruction and generation of virtual worlds

The last step of the proposed system is the 3D reconstruction of the table ball and ball movements. To achieve this, the output provided by the MOLT algorithm is used along with the information provided in Section 3.1.1. However, generating a 3D virtual world depends largely on the mechanics and syntax of the selected 3D language. For this reason, there are two main parts of the generation: the objects (table and balls) and the animations of the objects (ball movements). To achieve this goal, we selected the open X3D standard [68], which defines a language to generate 3D objects and worlds that can be visualized and shared on web pages. The steps to perform the 3D reconstruction are detailed in Procedure 8, which receives the size of the table (*baize\_size*, *pocket\_size*) and balls (*ball\_size*) as well as ball colors and movements (*ball\_colors*, *balls\_tracking*):

1. Gen\_Table: Table generation has two different cases: tables without pockets, as in the case of carom billiards, and tables with pockets, as in the case of blackball

and snooker. In the first case (*pocket\_size* == 0), the baize and table 3D object generation are quite simple, with rectangular objects for the table rails and a rectangular plane for the baize with the size specified by *baize\_size*. However, in the second case (*pocket\_size* > 0), table generation is a complex process because the pockets need to be erased from the table rails and the baize. Nevertheless, this problem can be simplified using the pocket diameter (*pocket\_size*) and the object “*Extrusion*” defined by the X3D standard. With the “*Extrusion*”, the shape of the object can be created by specifying a set of points in space. In the case of the baize, the corner pockets can be generated accounting for the radius of the pockets and from the vertices, taking points from 0° to 90° and points from 0° to 180° for the side pockets. The holes in the rails of the table are created following the same procedure.

2. Gen\_Balls: Ball generation is a simple task since it only takes the radius value and the color of every ball. In fact, it is only necessary to use the “*Sphere*” object defined by the standard.
3. Gen\_Animation: The last step in this procedure is animating ball movements. To achieve a simulation of the movement of each ball, the use is made of the balls and the table generated in the previous steps as well as the information provided by the MOLT algorithm. As the output of the MOLT algorithm (Procedure 3) is the position of each ball over time, it is possible to use the “*PositionInterpolator*” object defined by the

---

```

1: Inputs: (tracker_population, exploration_radius)
2: for each population of every ball do
3:    $bt_1 \leftarrow \text{tracker\_population}[1]$ 
4:    $bt_2 \leftarrow \text{tracker\_population}[2]$ 
5:    $bt_3 \leftarrow \text{tracker\_population}[3]$ 
6:   if  $bt_1.\text{Center\_point}.z \geq 0$  then
7:     for  $t \leftarrow 1, 2, 3, \dots, n$  do
8:       if  $t \leq n * 0.5$  then
9:          $\text{tracker\_population}[t].\text{Center\_point} \leftarrow \text{New\_Center}(bt_1, \text{exploration\_radius})$ 
10:      end if
11:      if  $t > n * 0.5$  and  $t \leq n * 0.80$  then
12:         $\text{tracker\_population}[t].\text{Center\_point} \leftarrow \text{New\_Center}(bt_2, \text{exploration\_radius})$ 
13:      end if
14:      if  $t > n * 0.80$  then
15:         $\text{tracker\_population}[t].\text{Center\_point} \leftarrow \text{New\_Center}(bt_3, \text{exploration\_radius})$ 
16:      end if
17:    end for
18:  end if
19: end for
20: Output tracker_population

```

---

**Procedure 7** Steps to generate a new population of trackers.

standard to generate the animation of each ball. The “*PositionInterpolator*” object has two main variables that are obtained from the MOLT algorithm:

- *Key*: Represents each point in time.
- *keyValue*: Represents the position of the object in each *Key*.

Finally, Fig. 4 shows an example of the reconstruction provided by Procedure 8 at a given point in time for each modality.

## 4 Results

This section shows the results of the MOLT algorithm performance and the proposed 3D reconstruction system. The evaluation of the MOLT algorithm is analyzed using different metrics detailed in the following subsections. However, 3D reconstruction is difficult to evaluate quantitatively with other methods. For this reason, the comparisons are performed qualitatively by gathering the opinions of users for the 3D reconstruction system.

Hence, this section is divided into six subsections: material and dataset information used to carry out the experiments, metrics for tracking performance evaluation,

analysis of the MOLT algorithm performance, comparison with other tracking methods, comparison with author’s proposals applied to billiards, and results of the 3D reconstruction and generation of virtual worlds.

### 4.1 Material and dataset information

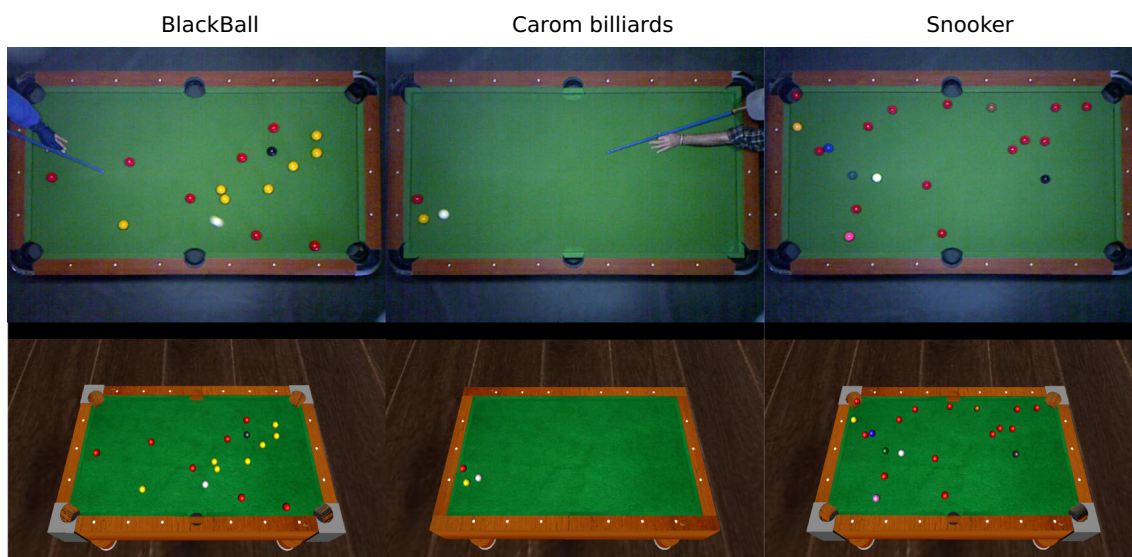
The results shown in the following subsections were obtained using the following computational equipment:

- Embedded device “Nvidia jetson nano” [69] with an ARM A57 @ 1.43 GHz and 4 GB LPDDR4.
- *RGB* sensor with the following specifications:
  - Focal length (in pixels): 525.
  - FOV (in degrees): 62.7.
  - Resolution (in pixels): 640 × 480.
  - Frame rate: 20 fps (frames per second).
  - Pixel correspondence (in mm): 3.47 mm (camera placed at 1853 mm over the table).

Additionally, since this work proposes a system able to work with three billiard modalities (blackball, carom billiards and snooker), the following billiard elements are used to obtain the results:

- 
- 1: **Inputs:** (*baize\_size*, *pocket\_size*, *ball\_colors*, *ball\_size*, *balls\_tracking*)
  - 2:  $table\_X3D \leftarrow Gen\_Table (baize\_size, pocket\_size)$
  - 3:  $balls\_X3D \leftarrow Gen\_Balls (ball\_colors, ball\_size)$
  - 4:  $animation\_file \leftarrow Gen\_Animation (table\_X3D, balls\_X3D, balls\_tracking)$
  - 5: **Output** *animation\_file*
- 

**Procedure 8** Steps to perform a reconstruction in a virtual world.



**Fig. 4** Real scene recorded by the camera and the 3D generated scene in a virtual world

- Billiard table with pockets with the following measurements:
  - Table size:  $211.5 \times 120.5 \times 78$  cm.
  - Baize and cushion area:  $185.4 \times 93.5$  cm.
- Pocket covers to make the table compatible with the carom billiards modality.
- Balls for blackball and carom billiard modalities of 57 mm diameter.
- Balls for snooker modality of 51 mm diameter.

Using the previously described elements, a dataset of 300 recorded billiard shots was generated, with 100 videos for each modality. These recordings are composed of more than 85,000 images. Additionally, a small sample was taken from these recordings, and the trajectories of all the balls were manually selected frame by frame, thus composing “ground truth” trajectories for quantitative comparisons of the errors. A total of 21, 16, and 16 manual ground truth trajectories for blackball, carom billiards, and snooker were generated, respectively. These trajectories stored as “.png” images are available online [70] along with the results that are described in the following subsections. This smaller dataset will be used in the following sections to test the performance of the proposed system and to compare the results with other algorithms in the tracking task.

## 4.2 Metrics for tracking performance evaluation

The proposal of the MOLT algorithm is one of the objectives of the work, and it is also one of the important tasks of the whole system. Thus, the trajectories provided by the method must be evaluated to check its performance. This evaluation is carried out by applying several metrics that are widely used in the scientific literature. Specifically, in this work, the Jaccard index [71, 72], IDF1 [73], MOTA [74], and MOTP [75] metrics are used:

- The Jaccard index [71, 72], also known as the *intersection over the union* coefficient, measures the similarity degree of two mathematical sets, in this case, the trajectory obtained by the tracking algorithm and the trajectory obtained manually (ground truth). This metric ranges from 0 to 1, representing 1 as the most accurate result.
- The IDF1 [73] metric computes a determination of which trajectories provided by the algorithm under evaluation are present in the ground truth trajectory in terms of proper association of the predicted ball path to the correct ball in the ground truth trajectory. IDF1 is usually used as a second-level metric because it focuses largely on association accuracy instead of detection accuracy. The best result for IDF1 output is 1.
- MOTA [74], which stands for multibject tracking accuracy, determines whether the paths are spatially similar to the ground truth for each frame that temporally composes the trajectory. It defines every tracking point as true positive, false negative, or false positive, according to the correct or incorrect identification of every tracker. This metric ranges from  $-\infty$  to 1, with 1 being the most accurate result.
- MOTP [75], which is the acronym for multiobject tracking precision, provides the accuracy of the spatial localization of the paths. It determines the intersection of the similarity of the trajectories with the set of true positive detections. It relies largely on the correct selection of the threshold value, and thus, it is very sensitive and inflexible. The MOTP output ranges from 0 to  $+\infty$ , with 0 being the most precise result.

The MOTA and MOTP metrics should be considered in combination, as MOTA measures the accuracy and MOTP the precision. Therefore, good tracking should account for both a good match of the obtained trajectory with the ground truth trajectory and a precise match of each ball. If MOTA is close to 1 and MOTP is high (not close to 0), the method can provide a good overall similar path to the ground truth, but the balls cannot be well detected individually. In contrast, if MOTP is close to 0 and MOTA is negative, this represents that the method can place precisely the ball in the exact position, but in most frames of the path, balls have not been located, and therefore, the trajectory is not accurate.

## 4.3 Analysis of MOLT algorithm performance

This subsection analyzes the accuracy of the proposed MOLT algorithm varying one of the most important parameters, the number of trackers. The aim of this analysis is to obtain the most efficient set of parameters for the MOLT algorithm. Specifically, we are focused on the minimum number of trackers that provide the best results. To analyze the effect of this parameter, it has gradually increased from 10 to 2000 in increments of 10. Additionally, the exploration radius assigned to each ball was fixed at 100 pixels for the cue ball and 80 pixels for the remaining balls in all experiments. The radius in the cue ball slightly increased because this ball normally moves at a higher speed than the other balls in all modalities.

The above process allows us to analyze the accuracy when the number of trackers is increased. To test the accuracy, among all the metrics described in the previous section, the Jaccard index was used because of its clear robustness and interpretability. This index compares the similarity of each result of the algorithm with the manual trajectories, as previously described. The results of these

experiments are summarized graphically in Fig. 5. Additionally, in the same figure, a boxplot representing the mean and the deviation with respect to the Jaccard index of all experiments is shown.

As shown in Fig. 5, in all the analyzed modalities, as the number of trackers increases, the Jaccard index increases, returning fewer errors compared with the manually selected trajectories. In fact, it can be observed that the limit is close to 0.9 in the Jaccard index for our proposed MOLT algorithm when the number of trackers per ball is close to 2,000. However, it can be seen that to achieve accurate tracking, such a large number of trackers is not necessary, and with values closer to 800, the results are quite precise. It should be noted that these results are obtained using the previously described materials, the capturing sensor does not have a high resolution or frame rate, and a difference of only one pixel between the tracked and manual trajectories decreases the Jaccard index. Additionally, the boxplot in the same figure shows that the outliers are related to a lower number of trackers per ball, and all the occurrences correspond to fewer than 200 trackers. In these cases, the balls cannot be tracked correctly because there are not enough trackers to cover the exploration radius (100 for the cue ball and 80 for all the other balls). In these experiments, it can be seen that the snooker modality is the most complicated to track because the balls are of a smaller diameter than those used in other modalities. Additionally, the maximum number of balls on the baize is 22, which is a high number of objects to track, and the deviations of one or two pixels from the manual trajectories significantly penalize the Jaccard index.

Another relevant issue is the occlusion problem. Due to the use of the tracker populations and diversity-oriented approach in generating these populations, the MOLT algorithm can track objects partially occluded by the cue or player's body. Therefore, our proposal is robust to occlusion, as shown in Fig. 7 in Appendix A for different cases.

Once the effect of the number of trackers is analyzed, the following subsections in this work will establish 2,000

trackers for the cue ball and 400 for all other balls. This selection of parameters is based on the premise that the cue ball is the fastest ball in the images and the other balls do not need a large number of trackers to ensure Jaccard index values higher than 0.7, as shown in Fig. 5.

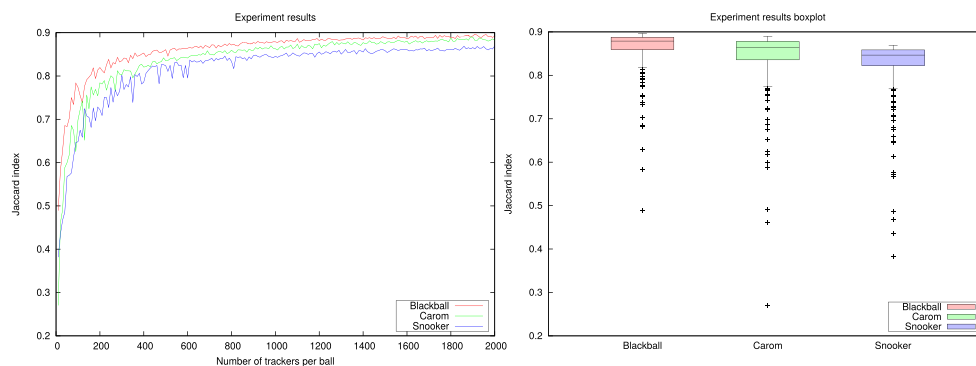
#### 4.4 Comparison with other tracking methods

In this subsection, the MOLT algorithm is compared with seven tracking methods that are designed for general object tracking [76]. The selected algorithms to perform the comparisons are listed below:

- Boosting [77].
- Multi-Instance Learning - MIL [78].
- Kernelized Correlation Filters - KCF [79].
- Tracking, Learning, and Detection - TLD [80].
- Median Flow [81].
- Minimum Output Sum of Squared Error - MOSSE [82].
- Discriminative Correlation Filter with Channel and Spatial Reliability - CSRT [83].

The above tracking methods are compared in Table 2 with the MOLT algorithm. To ensure a fair comparison between all the algorithms, the same images and the same number of trackers in the MOLT algorithm were used. The parameters of the MOLT algorithm for all the experiments are fixed those detailed at the end of Section 4.3. Thus, in all the following experiments, the executions were run using the same set of parameters. Moreover, for each recording, the tracked trajectories retrieved by each algorithm, as well as rendered videos comparing the results with the results provided by the proposed MOLT method, are available in the dataset in a sub-folder titled *TrackingResults*.

As can be observed in Table 2, in general, all the methods can identify the balls, as the IDF1 metric shows, with the exception of the TLD algorithm, where all the considered metrics exhibit poor performance. In contrast, the proposed MOLT algorithm achieves the most accurate and precise results in each metric for any billiard modality in general. The only case in which the MOLT algorithm



**Fig. 5** Jaccard index results for the increase of the number of trackers parameter

**Table 2** Comparison of different tracking methods and proposed MOLT algorithm

Modality	Method	Average IDF1	Average Jaccard	Average MOTA	Average MOTP
BlackBall	KCF	0.9377	0.2555	0.8813	1.8404
	TLD	0.2763	0.1781	-0.4039	540.0368
	MIL	0.9446	0.3915	0.8900	16.5180
	BOOSTING	0.9372	0.3792	0.9353	4.0346
	MOOSE	0.9377	0.2208	0.8813	1.8404
	CSRT	<b>1.0000</b>	0.5004	<b>1.0000</b>	1.5353
	MEDIAN FLOW	<b>1.0000</b>	0.7091	<b>1.0000</b>	0.9935
	MOLT	<b>1.0000</b>	<b>0.8771</b>	<b>1.0000</b>	<b>0.3466</b>
Carom billiards	KCF	0.7166	0.04650	0.4331	<b>2.1709</b>
	TLD	0.5510	0.2701	0.1020	607.2757
	MIL	0.4785	0.3145	-0.0431	146.2654
	BOOSTING	0.7166	0.1196	0.4331	2.4304
	MOOSE	0.5102	0.03875	0.0204	265.1689
	CSRT	0.7166	0.3110	0.4331	2.6266
	MEDIAN FLOW	0.9909	0.5849	0.9819	16.4874
	MOLT	<b>1.0000</b>	<b>0.8751</b>	<b>1.0000</b>	3.0680
Snooker	KCF	0.8431	0.2174	0.7587	15.6179
	TLD	0.1060	0.1314	-0.7649	469.8335
	MIL	0.7936	0.3564	0.6864	69.2982
	BOOSTING	0.8778	0.3084	0.7725	17.3635
	MOOSE	0.8429	0.4429	0.7391	4.4377
	CSRT	0.8882	0.3884	0.8654	1.9956
	MEDIAN FLOW	0.9360	0.6313	0.8721	4.1745
	MOLT	<b>1.0000</b>	<b>0.8578</b>	<b>1.0000</b>	<b>0.9537</b>

In **boldface**, best results for each metric and method.

is not the best is in the carom billiards modality for the average MOTP metric, in which the KCF, boosting, and CSRT algorithms slightly outperform the MOLT method. However, the MOLT method is both very accurate in obtaining the trajectory most similar to the ground truth trajectory, and very precise in locating the object along that trajectory. For instance, in the carom billiards modality, KCF detected precisely (MOTP: 2.1709) the balls but had a very poor response in providing a trajectory that matched the ground truth trajectory (MOTA: 0.4331).

In addition, according to the results of the Jaccard index, as Table 2 shows, the values obtained for the different modalities by the KCF, TLD, and MIL algorithms are, in general, remarkably inaccurate. This poor performance may be because these algorithms cannot track such small objects in an image from which they cannot obtain features other than their edge and color. For this reason, when there is minimal ball movement, they cannot track them accurately. Additionally, with these algorithms (especially in the case of TLD), once a tracked object is lost, the algorithms cannot

track correctly in subsequent frames. Another issue is that in the carom billiards modality, the results are even less accurate. This behavior is directly related to the limited number of balls to be tracked on the baize (three balls). Although it may be thought that a larger number of balls would result in greater errors, the opposite is actually true. This is because, in these modalities, the majority of balls are not in movement during a shot, and the algorithms consider the stationary balls to be perfectly tracked, thus increasing the accuracy. However, even if more balls are in motion during blackball and snooker modalities, the algorithms cannot accurately track the balls. For this reason, the Jaccard results are stricter and lower, as a few pixels deviate from the objects' trajectory penalizes its value greatly in contrast to the other metrics.

As a result of the values shown in Table 2, it can be seen that in general, the most accurate algorithm considering the Jaccard index is the proposed MOLT method, which achieves high and similar values of approximately 0.86. In fact, the correct functionality of the MOLT algorithm is



verified because it achieves in general the best values for each metric for all three modalities.

Finally, visual samples of the *intersection over the union* of the tracker algorithm results and manual results are available for each game modality in the dataset and a reduced number of cases are shown in Figs. 8, 9, and 10 of Appendix B.

#### 4.5 Comparison with other authors' proposals applied to billiards

To date, several comparisons using different metrics have been carried out with general object tracking algorithms. However, as Section 2 describes, other authors have proposed ideas for tracking balls in different billiard modalities. Hence, this section compares the results obtained by the MOLT algorithm using the same metrics as in the previous section with the methods proposed by Vachaspati and Legg et al. The reason for selecting these works to compare with the proposed MOLT algorithm is based on the following considerations:

- Their methods are employed in a real environment with real billiard tables and not prototyping tables with reduced scales.
- Their methods use RGB sensors without the need to use depth information to track objects.
- Their methods identify and track the different balls and are not limited to tracking just the cue ball or a small number of balls.

To perform a fair comparison, the set of parameters of the MOLT algorithm are the same as those selected in previous subsections, so the results are the same, as shown in Table 2. For the Vachaspati and Legg et al. methods, the parameters have been selected according to the suggestions provided by

the respective authors for best performance. Additionally, the experiments are carried out by applying the same subset of prerecorded videos analyzed in the previous subsections.

Table 3 shows the comparison between Vachaspati, Legg et al., and the MOLT proposals. It can be observed that the methods proposed by Vachaspati and Legg et al. obtain more precise results than the tracking algorithms analyzed in the previous subsection. The method proposed by Legg et al. outperforms the method proposed by Vachaspati for all the modalities in the MOTP metric and in the Jaccard index but not in the carom billiards modality in this last metric. These results are due to the steps of the method followed by Vachaspati, which is based on circle detection, and in the carom billiards modality, the speed of the cue ball and the strikes with the yellow and the red ball are slower than in the other modalities. Hence, the accuracy of the Vachaspati method for MOTA and the Jaccard index, when the ball movements are slow and are not blurred, is increased. These blurred movements are frequently obtained when the frame rate of the capturing sensor is low. Examples of blurred balls in movement are shown in Fig. 6.

Regarding the comparison of the proposed MOLT algorithm, the results show that in cases of blurred movements, the MOLT algorithm is robust, unlike the proposals of Vachaspati and Legg et al. This is because the MOLT algorithm, as explained in Section 3.2, tracks the balls based on the similarity to the initial state of the balls, in contrast with circle detection of the Vachaspati proposal and the specular highlight brightness detection of the proposal of Legg et al.

Moreover, the values of the metrics shown in Table 3 are not only an effect of the frame rate of the capturing sensor but also of low-resolution sensors. In fact, when a ball is not moving, the exact center position of the ball is not simple to obtain because the pixels and brightness of the ball captured by low-resolution sensors change in every

**Table 3** Comparison with different authors' proposals

Modality	Method	Average IDF1	Average Jaccard	Average MOTA	Average MOTP
BlackBall	Vachaspati	<b>1.0000</b>	0.7157	<b>1.0000</b>	3.3439
	Legg et al.	0.9738	0.7323	0.9475	3.2321
	MOLT	<b>1.0000</b>	<b>0.8771</b>	<b>1.0000</b>	<b>0.3466</b>
Carom billiards	Vachaspati	0.9705	0.7543	0.9410	16.0386
	Legg et al.	0.9433	0.6736	0.8866	12.4471
	MOLT	<b>1.0000</b>	<b>0.8751</b>	<b>1.0000</b>	<b>3.0680</b>
Snooker	Vachaspati	0.9594	0.6723	0.9193	5.1988
	Legg et al.	0.9194	0.7338	0.8389	3.3145
	MOLT	<b>1.0000</b>	<b>0.8578</b>	<b>1.0000</b>	<b>0.9537</b>

In **boldface**, best results for each metric and method in each billiard modality.

frame, generating errors. For this reason, the Vachaspati and Legg et al. methods are more sensitive and cannot accurately track static objects. This problem is mitigated in the MOLT algorithm because the tracker population of each ball and the diversity-oriented approach of the generation of new populations helps to avoid these errors.

For these two main reasons, the proposed MOLT algorithm provides the most accurate results according to all the considered metrics. For instance, MOLT obtains Jaccard index values that are 0.12 higher than those obtained by the other methods in all modalities. A similar situation occurs with the IDF1 and MOTA metrics, where it is shown that the best possible results are obtained. Finally, in the case of the MOTP metric, it is shown that the MOLT result is more precise, and this fact is magnified in the billiard modality with differences greater than 9 in the values obtained.

Therefore, it is proven that MOLT is the most precise method with low-frame-rate and low-resolution sensors. Also, in order to provide visual results of the comparisons, the *intersection over the union* of Vachaspati, Legg et al. and MOLT algorithms with the manual trajectories are available for each modality in the dataset and a reduced number of cases are shown in Figs. 11, 12, and 13 of Appendix C.

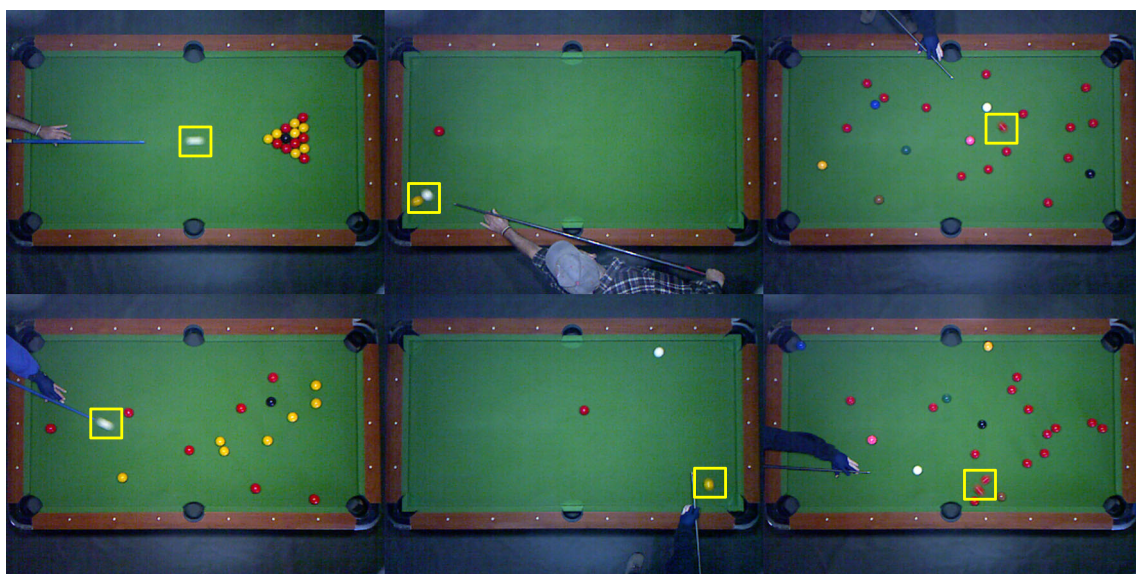
#### 4.6 Results of the 3D reconstruction and generation of virtual worlds

The previous sections analyzed the effect of using different tracking algorithms. These algorithms are necessary to perform the 3D reconstruction, so in this section, the degree of acceptance of the developed system by various users is analyzed. To achieve these qualitative results, the opinions of expert users were obtained through the *mean opinion*

*score (MOS)* procedure [84]. The MOS procedure was carried out with ten users who usually play billiards. This number of users was selected according to ITU-T Rec. P.911 [85], which standardizes the use of MOS and specifies that the number of experts used in the assessment should be above 6. The users reported their opinion regarding fluency, quality of reconstruction, and possible usefulness using questionnaires. Users provided a response for each feature ranging from 0, the worst possible result, to 10, the best possible result.

The procedure to gather the opinions of the different users was as follows:

1. First stage: Users used the proposed system through a developed interface (accompanied by a user manual) to capture images and obtain the results of the reconstructions in a virtual world. In this way, the user had real-world experience at the gaming table to compare with the result obtained in the virtual world.
2. Second stage: Each user was provided with a list of different 3D reconstructed shots. The shots used in the MOS procedure were the small subset used in the previous section mixed with the remaining reconstruction of the dataset. In this way, each user was asked to express his or her opinion regarding the abovementioned scale for each reconstruction subset. In addition, users were informed that the reconstruction could be repeated, but they would not be informed which of them would be repeated nor would they be numbered. This prevents a user from influencing his or her opinion regarding the previous results, thus avoiding favoring some results over others. In total, each case was presented twice to the users. Once the two opinions



**Fig. 6** Examples of blurred balls in movement

of each case were collected, the average opinion of each user was calculated.

The generated reconstruction, as well as rendered videos, are available online for each recording of the dataset in a subfolder titled *animationResults*. In this sub-folder, not only the reconstruction of the proposed system but also the reconstructions of the results of the methods of Vachaspati and Legg et al. were generated and compared in the previous section. Note that the different users did not see the results of these authors and only evaluated the results of our proposed system. The results of the reconstructions are shown in Figs. 14, 15, and 16 in Appendix D for each modality. In these figures a given point in time of the reconstruction and the real world image of that same moment are shown in order to appreciate the degree of similarity.

The opinions of the users for each modality are summarized in Table 4. As can be observed in the overall mark of each modality, the opinion of the users is related to the number of balls; the higher the number of balls, the lower the overall mark. These results are caused by a jittering effect of the 3D-generated balls. This effect is due to the inaccuracy of the MOLT algorithm, despite it being the most accurate of the analyzed tracking algorithms. Any failure of one pixel in the tracking corresponds to 3.47 mm in the real world. For this reason, when the algorithm returns only two pixels of difference with respect to the real ball position, jittering can be seen in the animation. Thus, a higher number of balls increases the visual perception of balls suffering the jittering effect. Despite this fact, the average result provided by the users is greater than 7 (out of 10), for each modality. Both fluency and reconstruction quality obtain average marks of 8 and 7.9 (out of 10) respectively. Finally, learning utility presents lower values than the other evaluated aspects, providing an average value of 6.8 (out of 10) for the three modalities. This shows that the proposed system has the potential to be used as a learning system or for entertainment purposes.

## 5 Conclusions

Billiards considering the multiple modalities, is a sport widely practiced around the world. There are several pro-

posals for the applicability of virtual reality or augmented reality systems to improve the skills of novice or amateur players. The reconstruction of shots in virtual scenarios based on computer vision tracking algorithms is an example of such a system. However, when using low-quality, low-frame-rate devices, the tracking algorithms have flaws in the tracking accuracy of objects that appear blurred in the image or with poorly defined edges. Moreover, considering billiard modalities such as blackball or snooker, the problem becomes complex due to the multiple identically colored objects to be tracked in such unfavorable conditions.

In this context, this paper presents two main contributions. The first main contribution is a new “*multiobject local tracking (MOLT) algorithm*” to perform the task of tracking ball movements. The MOLT algorithm is designed to track multiple small objects robustly and operate in unfavorable conditions for tracking where image-capturing devices have low resolution and frame rates. The second contribution is a whole system capable of performing a 3D reconstruction in a virtual world of shots, collisions and ball movements. To carry out this second aim, the MOLT results are incorporated along with other preprocessing and postprocessing steps including the following: segmentation of the balls, identification and classification of the different balls, 3D table generation, and reconstruction of the ball motion.

The proposed MOLT algorithm and the whole system were tested on three billiard modalities: blackball, carom billiards and snooker. For each modality, 100 recordings along with the reconstruction results and the outputs of the MOLT algorithm are available online to facilitate future comparisons. In particular, the MOLT algorithm is compared with the results obtained by nine other methods: seven general object tracking methods and two methods proposed by other authors designed for billiard tracking. From the experiments performed, it is observed that the MOLT algorithm achieves in the majority of the cases the most accurate and precise results for all the experiments analyzed considering the IDF1, MOTA and MOTP metrics. Moreover, considering the Jaccard index, which is one of the most interpretable metrics, the MOLT algorithm outperforms the other methods, obtaining the highest scores with values above 0.85 for each billiard modality.

With regard to the complete system, the 3D reconstructions of the ball movements were evaluated by collecting

**Table 4** Mean opinion score results for each modality

	Blackball	Carom billiards	Snooker	Average
Fluency	7.7	8.4	8	8
Reconstruction quality	8.1	8	7.6	7.9
Learning utility	7	6.8	6.8	6.8
Overall mark	7.6	7.7	7.4	7.6

the opinions of different users who regularly play billiards. They rated the 3D reconstructions obtained with an average score of 7.6 (out of 10) for all modalities. These results are due to the accuracy and precision of the MOLT algorithm incorporated into the system.

Enhancements to both the system as a whole and the MOLT algorithm are proposed for future work. Concerning the 3D reconstruction, animation of the cue can be incorporated into the reconstruction, as well as a virtual avatar controlling the cue. In addition, since the lowest user rating was on usefulness for learning, the incorporation of metrics such as shot degrees, ball acceleration, ball speed, and frame score is proposed. In terms of reconstruction quality,

a direct consequence of the output of the MOLT algorithm, the incorporation of a new postprocessing module is proposed to correct for the small effect of ball jittering. Finally, given the accurate results of the MOLT algorithm, future research can focus on the use of this algorithm in other environments, such as tracking people [86] in a shopping mall or on the street.

## Appendix A: Examples of occlusions

This section shows the results of the MOLT algorithm when partial occlusions occurs.



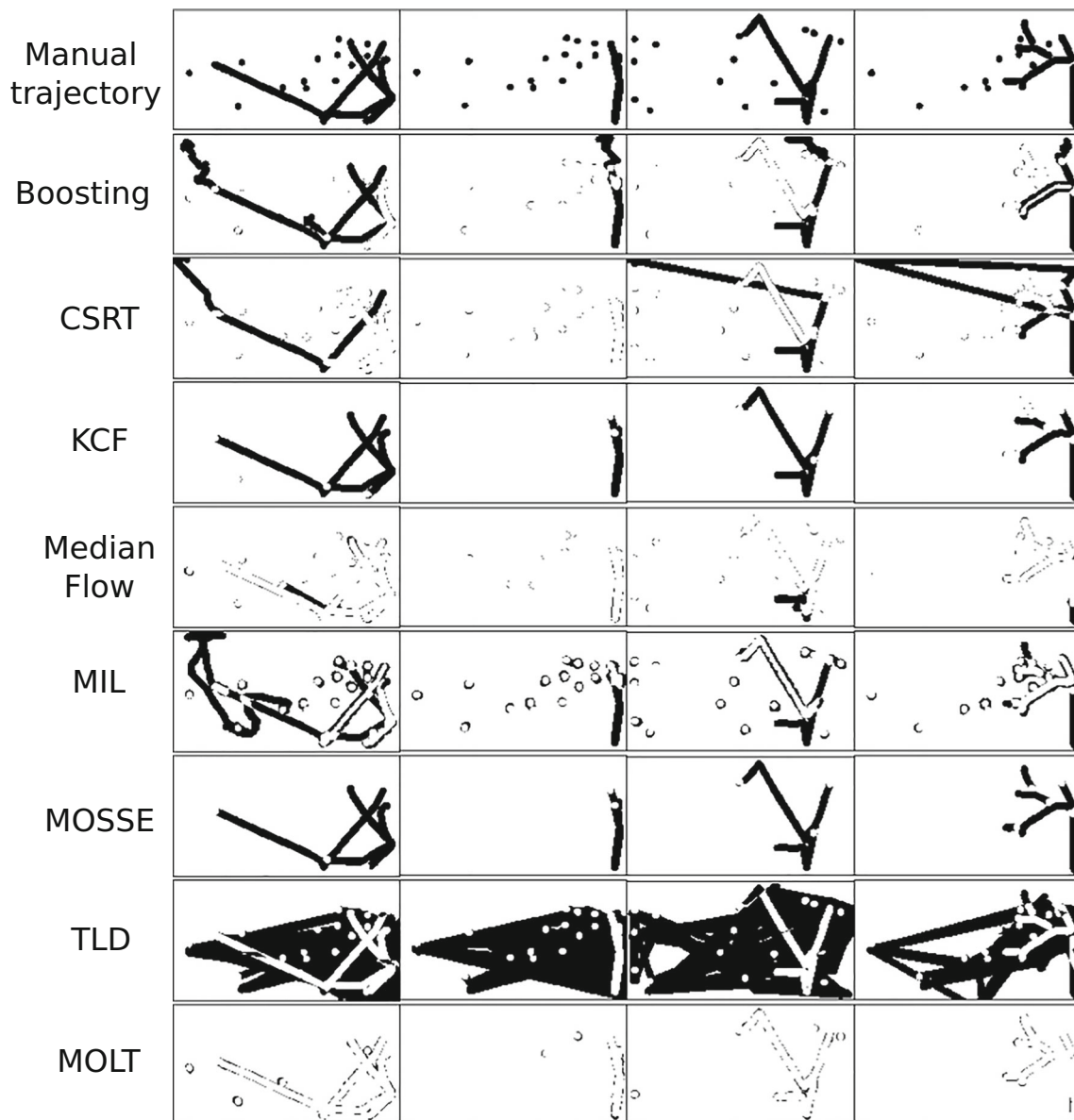
**Fig. 7** Examples of partial occlusion handled by the MOLT algorithm

## Appendix B: Visual results of the comparison with other tracking methods

show these results for each modality analyzed in this work:

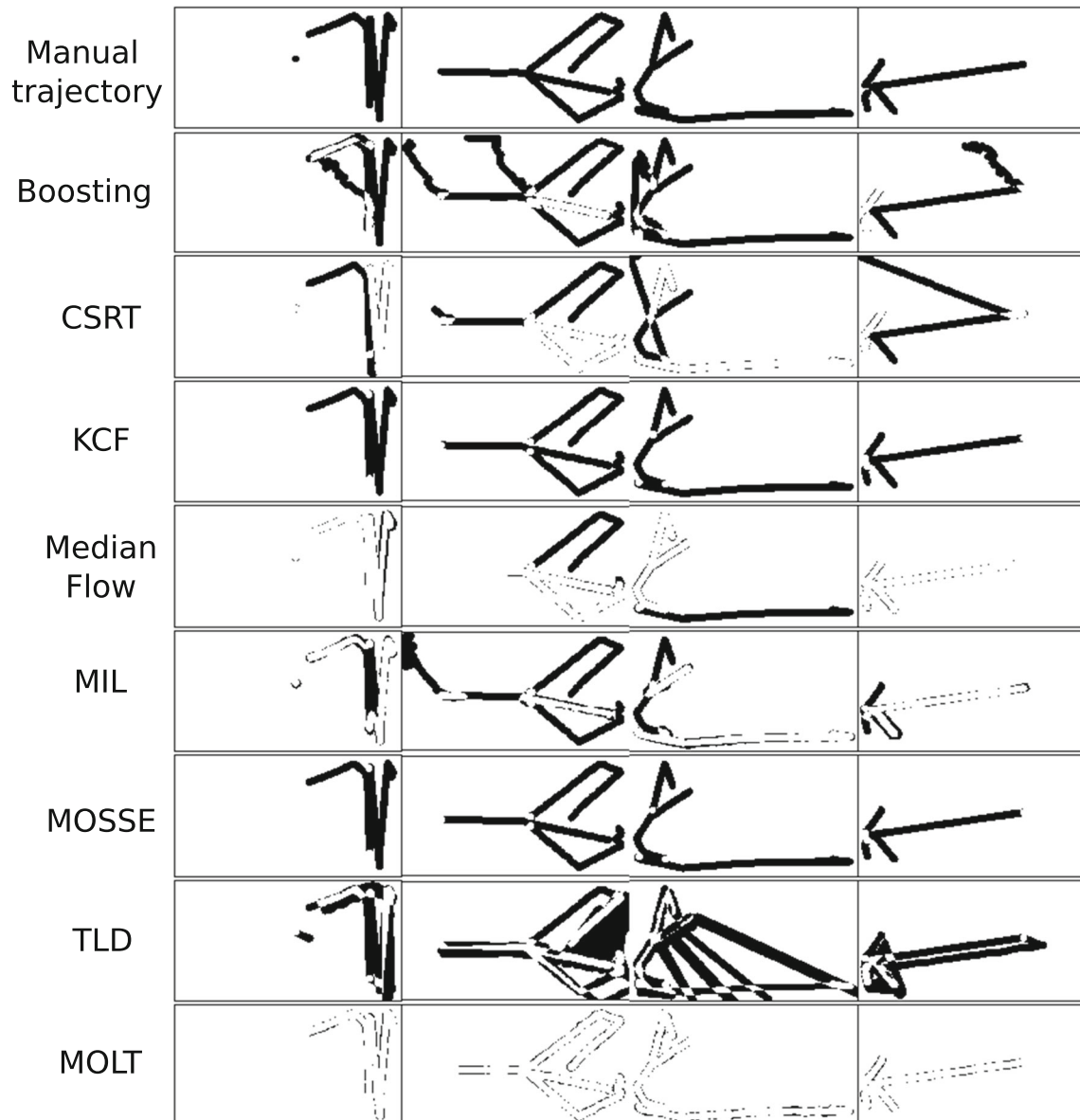
This appendix contains visual results of the comparison with other tracking methods. The following subsections

### B.1 Blackball modality



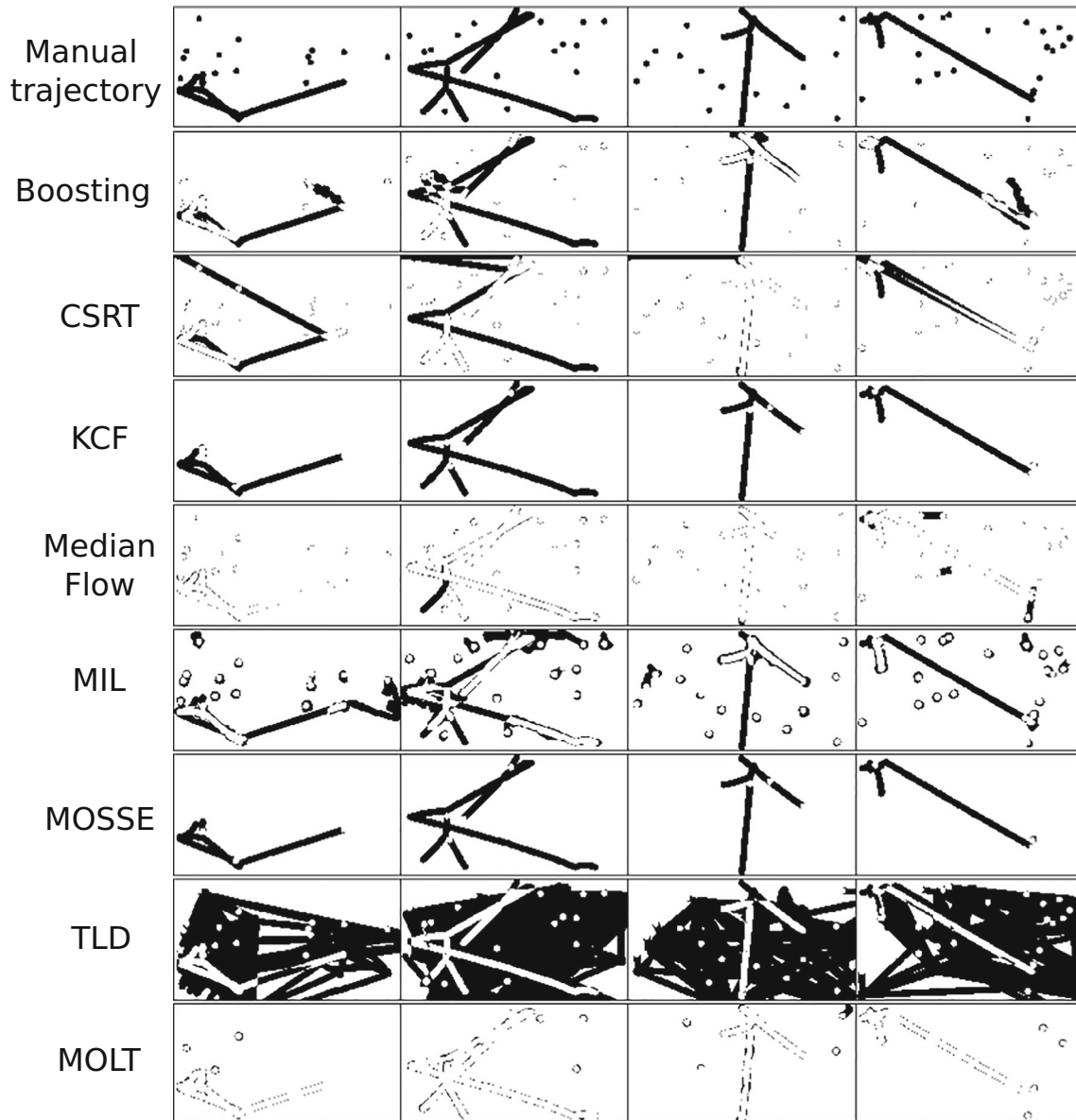
**Fig. 8** Manual trajectories, and the intersection over the union of each algorithm with the manual trajectory for different cases of the blackball modality. Note that less black color in images means more accurate tracking

## B.2 Carom billiard modality



**Fig. 9** Manual trajectories, and the intersection over the union of each algorithm with the manual trajectory for different cases of the carom billiard modality. Note that less black color in images means more accurate tracking

### B.3 Snooker modality

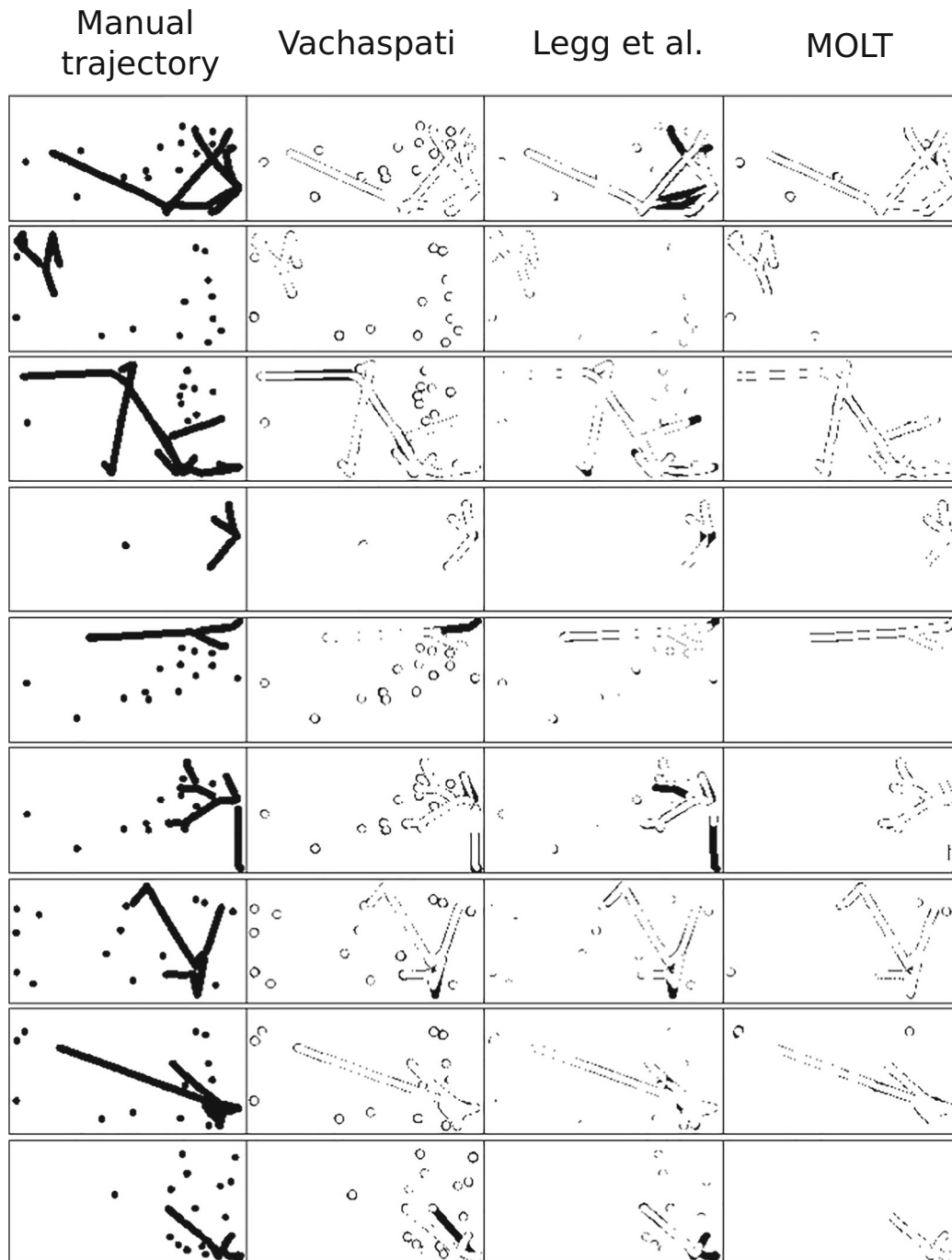


**Fig. 10** Manual trajectories, and the intersection over the union of each algorithm with the manual trajectory for different cases of the snooker modality. Note that less black color in images means more accurate tracking

## Appendix C: Visual results of the comparison with other authors' proposals applied to billiards

This appendix contains visual results of the comparison with other other authors' proposals. The following subsections show these results for each modality analyzed in this work:

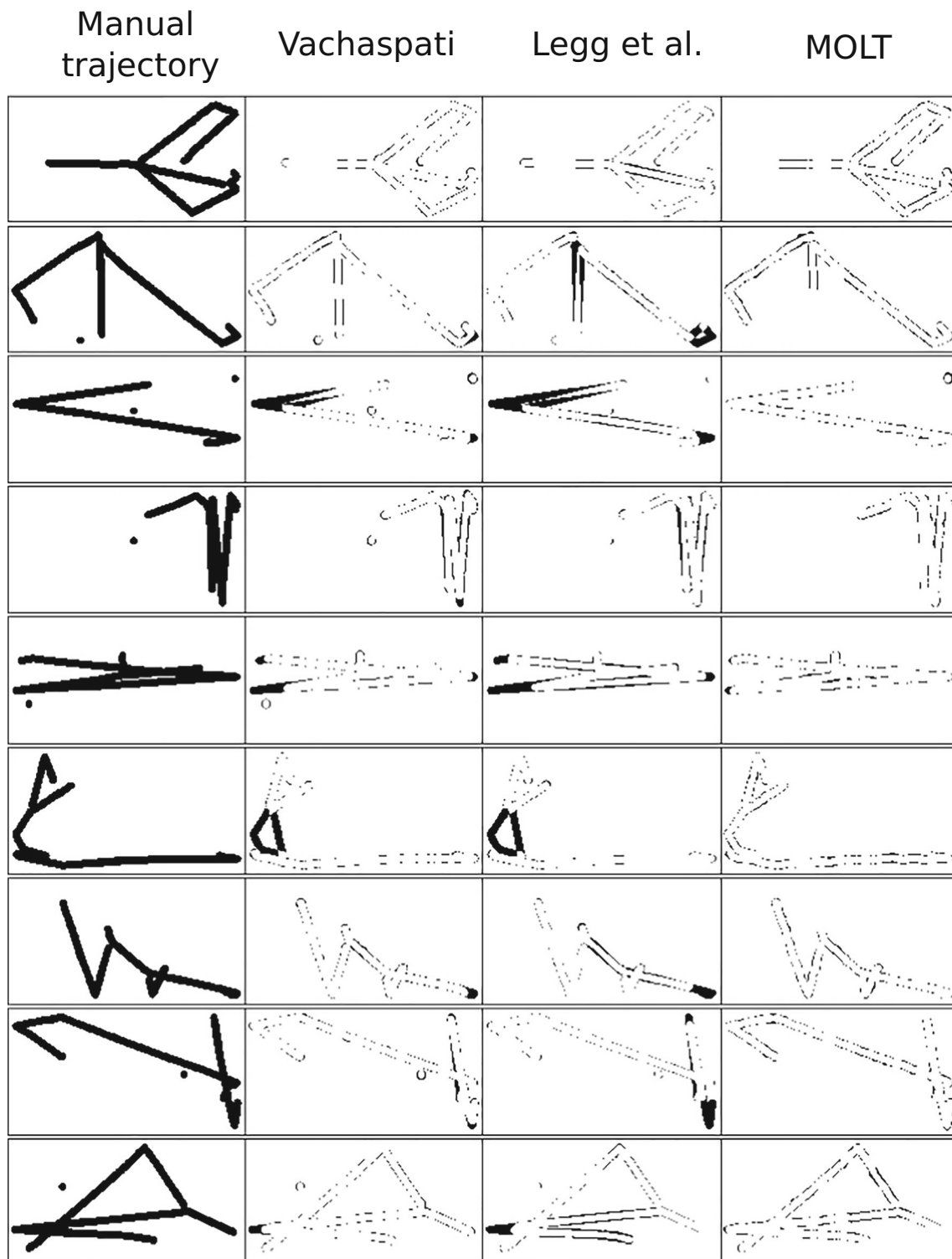
### C.1 Blackball modality



**Fig. 11** Manual trajectories, and the intersection over the union of each proposal with the manual trajectory for different cases of the blackball modality. Note that less black color in images means more accurate tracking

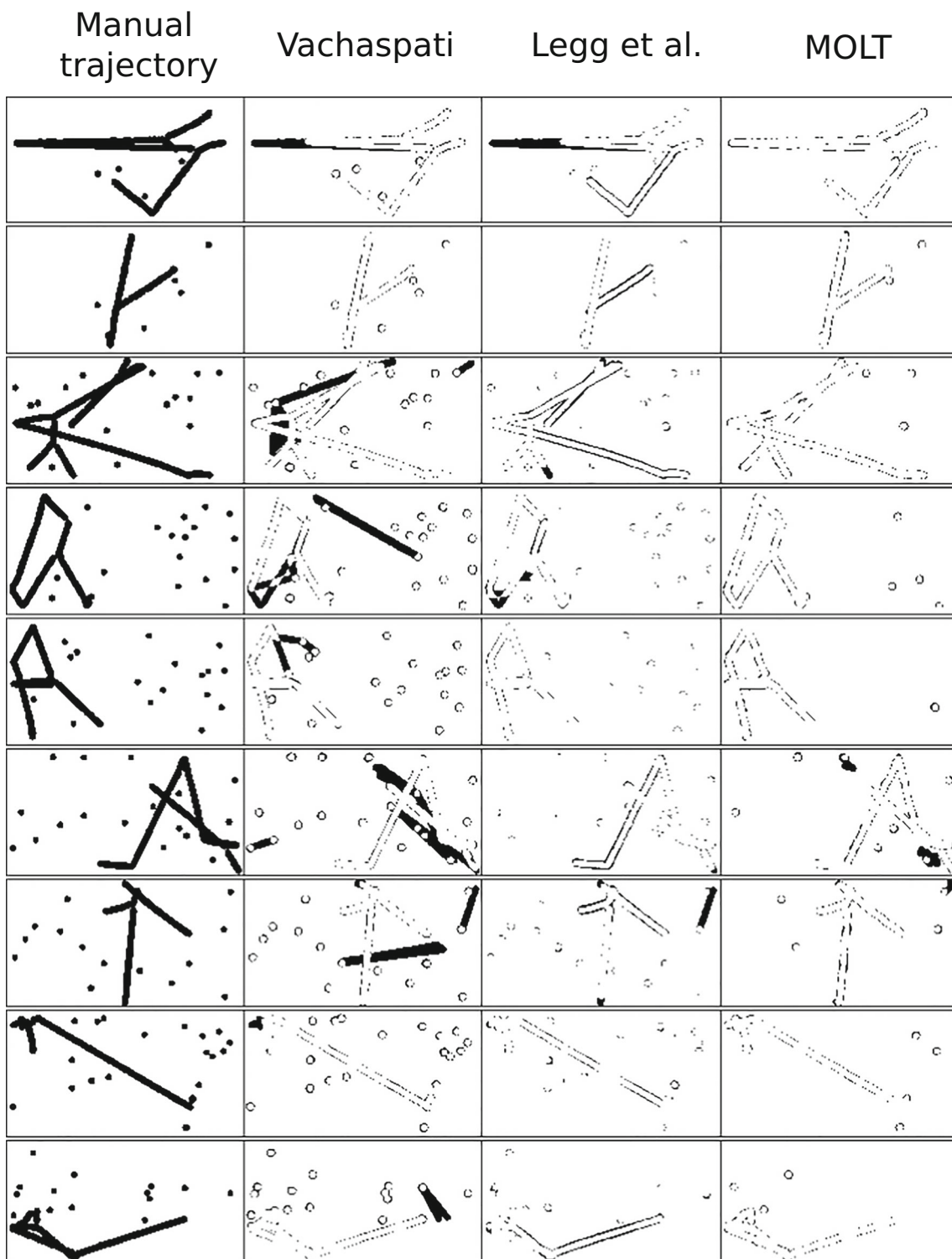


## C.2 Carom billiard modality



**Fig. 12** Manual trajectories, and the intersection over the union of each proposal with the manual trajectory for different cases of the carom billiard modality. Note that less black color in images means more accurate tracking

### C.3 Snooker modality



**Fig. 13** Manual trajectories, and the intersection over the union of each proposal with the manual trajectory for different cases of the snooker modality. Note that less black color in images means more accurate tracking

## Appendix D: Visual results of the 3D reconstruction

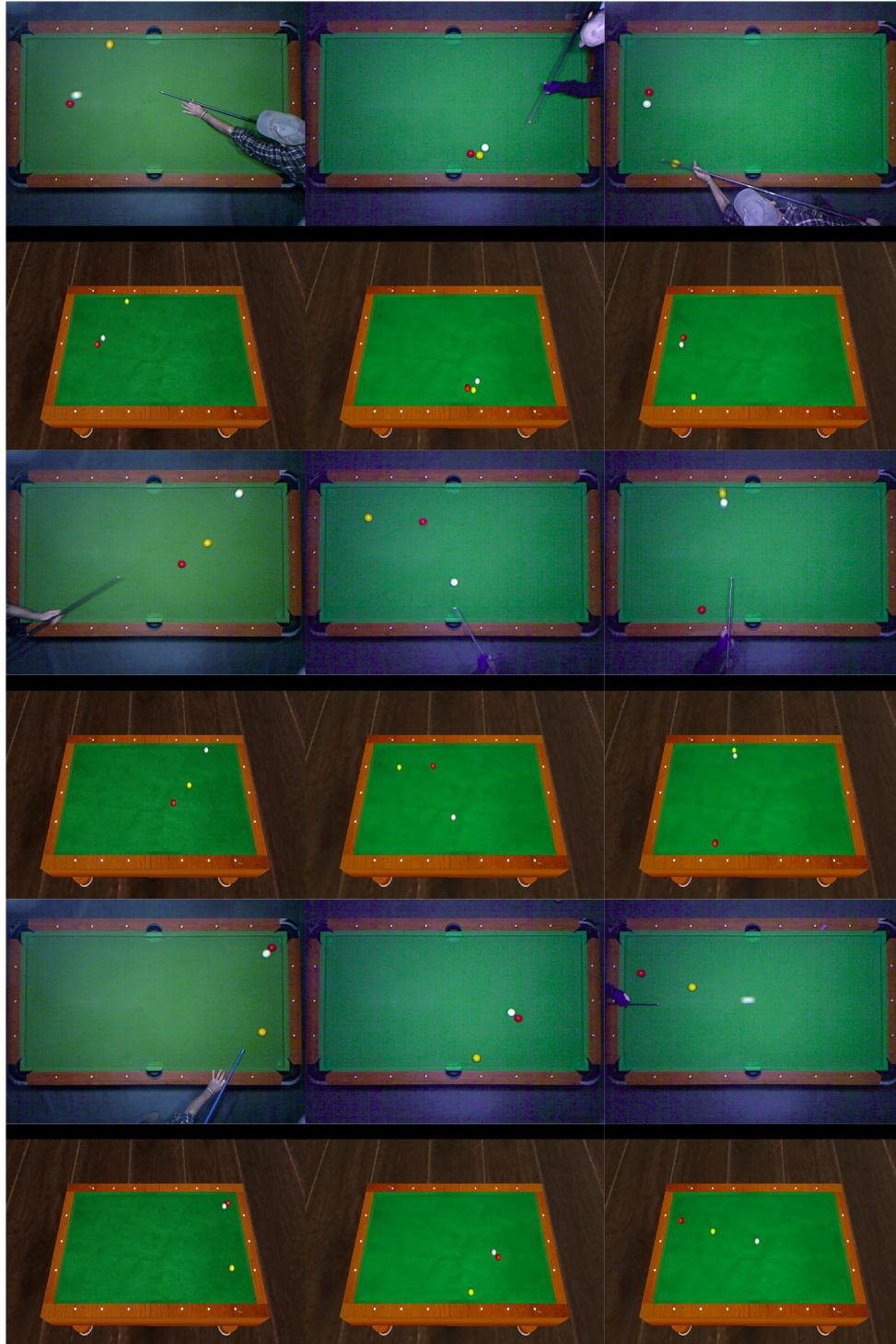
This appendix contains visual results of the proposed system and the main aim of this work, the 3D reconstruction of billiards shots. The following subsections show these results for each modality analyzed in this work:

### D.1 Blackball modality



Fig. 14 Real world scene and the 3D reconstructed virtual world at a given point in time for different shots in the blackball modality

## D.2 Carom billiard modality



**Fig. 15** Real world scene and the 3D reconstructed virtual world at a given point in time for different shots in the carom billiard modality

### D.3 Snooker modality



**Fig. 16** Real world scene and the 3D reconstructed virtual world at a given point in time for different shots in the snooker modality

**Acknowledgements** This research is supported by the MINECO project “*Intelligent distributed processing architectures in Fog level for the IoT paradigm (Smart-Fog)*” Ref: RTI2018-098371-B-I00, the “*Programa Operativo FEDER Andalucía 2014-2020*” Ref: 1380974-F, and the Advanced Informatics Research Group – GIIA (TIC-252) Universidad de Córdoba (Spain). Héctor Martínez is a POSTDOC\_21.00025 fellow supported by Junta de Andalucía (Spain). Thanks for Funding for open access charge: Universida de Córdoba/CBUA.

**Author Contributions** Not applicable.

**Funding** Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature. This research is supported by the MINECO project “*Intelligent distributed processing architectures in Fog level for the IoT paradigm (Smart-Fog)*” Ref: RTI2018-098371-B-I00 and the “*Programa Operativo FEDER Andalucía 2014-2020*” Ref: 1380974-F.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

**Availability of data and materials** The datasets used during the current study are available on the GitHub platform (<https://github.com/FJ-Rodriguez-Lozano/Billiard-dataset>).

**Code Availability** Not applicable.

## Declarations

**Ethics approval** Not applicable.

**Consent to participate** Not applicable.

**Consent for Publication** Participants were informed that the results of their opinions would be published in a way that their identity could not be revealed.

**Conflict of Interests** The authors of this paper declare that they have no conflict of interest.

## References

- Chen J, Zhenghao X, Ji J (2022) Multi-object tracking based on network flow model and orb feature. *Appl Intell*
- Luo W, Xing J, Milan A, Zhang X, Liu W, Kim T-K (2021) Multiple object tracking: a literature review. *Artif Intell* 293:103448
- Parihar A, Nagarkar P, Bhosale V, Desale K (2019) Survey on multiple objects tracking in video analytics. *Int J Comput Appl* 181(35):5–9
- Kim W, Moon S-W, Lee J, Nam D-W, Jung C (2018) Multiple player tracking in soccer videos: an adaptive multiscale sampling approach. *Multimedia Syst* 24(6):611–623
- Baysal S, Duygulu P (2016) Sentiscope: a soccer player tracking system using model field particles. *IEEE Trans Circuits Syst Video Technol* 26(7):1350–1362
- Kamble PR, Keskar AG, Bhurchandi KM (2017) Ball tracking in sports: a survey. *Artif Intell Rev* 52(3):1655–1705
- Li Z, Li X, Shi M, Song W, Zhao G, Yang R, Li S (2021) Tracking algorithm of snowboard target in intelligent system. *J Intell Fuzzy Syst* 40:3117–3125
- Moon S, Lee J, Nam D, Kim H, Kim W (2017) A comparative study on multi-object tracking methods for sports events. In: 2017 19Th international conference on advanced communication technology (ICACT), pp 883–885
- Dave Alciatore (2022) Glossary of pool and billiards terms and phrases. <https://billiards.colostate.edu/glossary/>
- Haar S, van Assel CM, Faisal AA (2020) Motor learning in real-world pool billiards. *Sci Rep* 10(1)
- Elmagd MA (2019) Is billiards considered a sport? *J Adv Sports Phys Educ* 02(07):157–161
- Supriadi B, Lesmono AD, Prastowo SHB, Prihandono T, Subiki Ridlo ZR, Rosyadi FA (2019) Study of central and non-central collisions in billiard games. *IOP Conf Ser: Earth Environ Sci* 243:012022
- Thai E, Kumar AR (2019) How haptic feedback in a mixed reality pool game affects real-life pool performance. *Proc Hum Factors Ergon Soc Annu Meet* 63(1):2323–2327
- World Confederation of Billiards Sports (2022) How to play Carom. <https://wcbs.sport/sports/how-to-play-carom/>
- The World Professional Billiards and Snooker Association (2022) Official rules of the game of snooker and english billiards. The World Professional Billiards & Snooker Association Limited (WPBSA)
- World Pool-Billiard Association (2016) The Rules of Play - Version 15.03.2016. <https://wppool.com/rules-of-play/>
- The World Pool-Billiard Association (WPA) (2018) Bring billiards as an additional sport to the Olympic Games Paris 2024. <https://wppool.com/bring-billiards-as-an-additional-sport-to-the-olympic-games-paris-2024-2/>
- Ling Y, Li S, Xu P, Zhou B (2012) The detection of multi-objective billiards in snooker game video. In: 2012 Third international conference on intelligent control and information processing, pp 594–596. IEEE
- Roerdink JBTM, Meijster A (2000) The watershed transform: Definitions, algorithms and parallelization strategies. *Fundam Inf* 41(1–2):187–228
- Legg PA, Parry ML, Chung DHS, Jiang RM, Morris A, Griffiths IW, Marshall D, Chen M (2011) Intelligent filtering by semantic importance for single-view 3d reconstruction from snooker video. In: 2011 18th IEEE international conference on image processing, pp 2385–2388. IEEE, Belgium
- Parry ML, Legg PA, Chung DHS, Griffiths IW, Chen M (2011) Hierarchical event selection for video storyboards with a case study on snooker video visualization. *IEEE Trans Vis Comput Graphics* 17(12):1747–1756
- Vachaspati P (2012) A computer vision system for 9-ball pool. Technical Report Massachusetts Institute of Technology
- Baekdahl BJ, Have S (2011) Detection and identification of pool balls using computer vision. PhD thesis, Vision, Graphics and Interactive Systems, Aalborg University
- Weatherford S (2013) Pool cue guide determination of guide vectors under adverse lighting, view aspect and scale. Technical Report, Department of Electrical Engineering, Stanford University, Stanford, United States

25. Uchiyama H, Saito H (2007) AR Display of visual aids for supporting pool games by online markerless tracking
26. Denman H, Rea N, Kokaram A (2003) Content-based analysis for video from snooker broadcasts. *Comput Vis Image Underst* 92(2–3):176–195
27. Hsu C-C, Tsai H-C, Chen H-T, Tsai W-J, Lee S-Y (2017) Computer-assisted billiard self-training using intelligent glasses. In: 2017 14th international symposium on pervasive systems, algorithms and networks & 2017 11th international conference on frontier of computer science and technology & 2017 third international symposium of creative computing (ISPAN-FCST-ISCC). IEEE, Exeter, Devon, United Kingdom
28. Larsen LB, Jensen RB, Jensen KL, Larsen S (2005) Development of an automatic pool trainer. In: Proceedings of the 2005 ACM SIGCHI International Conference on Advances in Computer Entertainment Technology, pp 83–87. ACM Press, Spain
29. Sousa L, Alves R, Rodrigues JMF (2016) Augmented reality system to assist inexperienced pool players. *Comput Vis Med* 2(2):183–193
30. Gao J, He Q, Zhan Z, Gao H (2016) Dynamic modeling based on fuzzy neural network for a billiard robot. In: 2016 IEEE 13th international conference on networking, sensing, and control (ICNSC), pp 1–4
31. Park J, Park J (2015) CAMshift algorithm based on geometric information for fast billiard ball tracking. In: 2015 12th international conference on ubiquitous robots and ambient intelligence (URAI), pp 137–138. IEEE, Korea
32. Pei L, Zhang H, Yang B (2021) Improved camshift object tracking algorithm in occluded scenes based on AKAZE and kalman. *Multimed Tools Appl* 81(2):2145–2159
33. Gao J, He Q, Gao H, Zhan Z, Wu Z (2018) Design of an efficient multi-objective recognition approach for 8-ball billiards vision system. *Kuwait journal of science* 45
34. Davies ER (2018) Chapter 11 - the generalized hough transform. In: Davies E. R. (ed) *Computer vision* (fifth edition), fifth edition edn., pp 299–339. Academic press
35. Alhaji R, Rokne J (eds) (2018) *Convolutional neural network*. Springer, NY
36. Cross R (2020) Impact force between two colliding billiard balls. *Phys Educ* 55(6):065002
37. Gabdulhakova A, Kropatsch WG (2014) Video analysis of a snooker footage based on a kinematic model. *Structural, Syntactic, and Statistical Pattern Recognition* 8621:223–232
38. Mathavan S, Jackson MR, Parkin RM (2009) Application of high-speed imaging to determine the dynamics of billiards. *Am J Phys* 77(9):788
39. Mathavan S, Jackson MR, Parkin RM (2016) Ball positioning in robotic billiards: a nonprehensile manipulation-based solution. *IEEE/ASME Trans Mechatronics* 21( 1):184–195
40. Tung KG, Wen Wang S, Tai WK, Lor Way D, Chang CC (2019) Toward human-like billiard ai bot based on backward induction and machine learning. In: 2019 IEEE Symposium series on computational intelligence (SSCI), pp 924–932
41. Bhagat KH (2018) *Automatic Snooker-Playing Robot with Speech Recognition Using Deep Learning*. PhD thesis, California State University, Long Beach ProQuest Dissertations Publishing, Degree Year 2018 10977867
42. Menini L, Possieri C, Tornambè A (2018) Tracking of a bouncing ball in a planar billiard through continuous-time approximations. *J Comput Nonlinear Dyn* 13:061006
43. Menini L, Possieri C, Tornambè A (2020) Trajectory tracking in rectangular billiards by unfolding the billiard table. *IFAC-PapersOnLine* 53(2):6195–6200. 21st IFAC World Congress
44. Menini L, Possieri C, Tornambè A (2021) Trajectory tracking of a bouncing ball in a triangular billiard by unfolding and folding the billiard table. *Int J Control* 0(0):1–14
45. Yu J, Huang Y, He Y (2018) Snooker video event detection using multimodal features. In: Proceedings of the 1st International Workshop on Multimedia Content Analysis in Sports. ACM, Seoul, Korea
46. Li S, Li B, Lu H, Xiao J (2021) Snooker match outcome prediction using ANN with inception structure. In: 2021 International conference on applications and techniques in cyber intelligence, pp 351–359. Springer, Fuyang, China
47. Sun Z, Chen J, Zhou H, Zhou D, Li L, Jiang M (2019) GraspSnooker: Automatic chinese commentary generation for snooker videos. In: Proceedings of the 28th International joint conference on artificial intelligence. International joint conferences on artificial intelligence organization, Macao, China
48. Chung DHS, Griffiths IW, Legg PA, Parry ML, Morris A, Chen M, Griffiths W, Thomas A (2014) Systematic snooker skills test to analyze player performance. *Int J Sports Sci Coaching* 9(5):1083–1105
49. Collingwood JAP, Wright M, Brooks RJ (2022) Evaluating the effectiveness of different player rating systems in predicting the results of professional snooker matches. *Eur J Oper Res* 296(3):1025–1035
50. Jebara T, Eyster C, Weaver J, Starner T, Pentland A (1997) Stochastic: augmenting the billiards experience with probabilistic vision and wearable computers. In: Digest of papers. First international symposium on wearable computers, pp 138–145. IEEE Comput. Soc, USA
51. Medved S (2020) *Augmented reality billiards assistant*. Williams Honors College, Honors Research Projects. 1104. [https://ideaexchange.uakron.edu/honors\\_research\\_projects/1104](https://ideaexchange.uakron.edu/honors_research_projects/1104)
52. Shih C (2010) Aiming strategy error analysis and verification of a billiard training system. *Knowl-Based Syst* 23(7):732–742
53. Shih C (2014) Analyzing and comparing shot planning strategies and their effects on the performance of an augmented reality based billiard training system. *Int J Inf Technol Decis Making* 13(03):521–565
54. Paolis LTD, Aloisio G, Pulimeno M (2009) A simulation of a billiards game based on marker detection. In: 2009 Second international conferences on advances in computer-human interactions. IEEE, Mexico
55. Mishima M, Suganuma A (2016) Development of stance correction system for billiard beginner player. In: 2016 International symposium on intelligent signal processing and communication systems (ISPACS), pp 1–5. IEEE, Thailand
56. Pinzon DMR, Gonzalez ECV, Gomez EJ (2017) Billiard game parameters calculation using a depth camera for augmented reality applications. *Contemp Eng Sci* 10:1171–1180
57. Wu F, Dellinger A (2017) Capturing reality for a billiards simulation. In: De paolis, LT, Bourdot, P, Mongelli, A (eds) *Augmented Reality, Virtual Reality and Computer Graphics*, pp 289–298, Springer, Cham
58. Kato J, Nakashima T, Takeoka H, Ogasawara K, Murao K, Shimokawa T, Sugimoto M (2013) Openpool: Community-based prototyping of digitally-augmented billiard table. In: 2013 IEEE 2nd global conference on consumer electronics (GCCE). IEEE, Japan
59. Dymora P, Mazurek M, Smalara K (2021) Modeling and fault tolerance analysis of zigbee protocol in iot networks. *Energies* 14(24)
60. OpenCV team (2022) *The OpenCV Reference Manual - 4.6.0*. <https://docs.opencv.org/4.6.0/>

61. Bugarinović Z, Pajewski L, Ristić A, Vrtunski M, Govedarica M, Borisov M (2020) On the introduction of canny operator in an advanced imaging algorithm for real-time detection of hyperbolas in ground-penetrating radar data. *Electronics* 9(3)
62. Min Allah N, Jan F, Alrashed S (2021) Pupil detection schemes in human eye: a review. *Multimed Syst* 27
63. Gonzalez RC, Woods RE (2018) Intensity transformations and spatial filtering. In: *Digital Image Processing*, 4th edn., pp 119–202. Addison-Wesley Longman Publishing Co., Inc. USA. Chap. 3
64. Lopez-Martinez A, Cuevas FJ (2019) Automatic circle detection on images using the teaching learning based optimization algorithm and gradient analysis. *Appl Intell* 49(5):2001–2016
65. Li C, Liu J, Wu Q, Bi L (2021) An adaptive enhancement method for low illumination color images. *Appl Intell* 51(1):202–222
66. Xu H, Ding C, Li P, Ji Y (2022) An active learning algorithm based on the distribution principle of bhattacharyya distance. *Mathematics* 10(11)
67. Thomas HC, Charles EL, Ronald LR, Stein C (2022) *Introduction to Algorithms*, Fourth Edition. The MIT Press, Cambridge
68. ISO/IEC 19775-1:2022 (2022) Information technology — Computer graphics, image processing and environmental data representation - Extensible 3D (X3D) - Part 1: Architecture and base components. Standard, International Organization for Standardization
69. NVIDIA Corporation (2021) Jetson Nano Developer Kit. <https://developer.nvidia.com/embedded/jetson-nano-developer-kit>
70. Rodríguez-Lozano FJ (2022) Billiard-dataset. <https://github.com/FJ-Rodriguez-Lozano/Billiard-dataset>
71. Rodríguez-Lozano FJ, León-García F, Ruiz de Adana M, Palomares JM, Olivares J (2019) Non-invasive forehead segmentation in thermographic imaging. *Sensors* 19(19)
72. Zheng X, Lei Q, Yao R, Gong Y, Yin Q (2018) Image segmentation based on adaptive k-means algorithm. *EURASIP Journal on Image and Video Processing* 2018(1)
73. Chen J, Xi Z, Wei C, Lu J, Niu Y, Li Z (2021) Multiple object tracking using edge multi-channel gradient model with orb feature. *IEEE Access* 9:2294–2309
74. Li J, Ding Y, Wei H-L, Zhang Y, Lin W (2022) Simpletrack: Rethinking and improving the jde approach for multi-object tracking. *Sensors* 22(15)
75. Rodrigues B, Scheid EJ, Willems J, Tornow M, Müller KOE, Stiller B (2022) Fusion data tracking system (fits). *IEEE Sensors J* 22(19):19060–19072
76. Dardagan N, Branin A, Džigal D, Akagic A (2021) Multiple object trackers in opencv: A benchmark. In: 2021 IEEE 30th international symposium on industrial electronics (ISIE), pp 1–6
77. Boosting algorithm (2021). In: Ikeuchi, K. (ed.) *Computer Vision*, pp 110–110. Springer
78. Fu C, Duan R, Kayacan E (2019) Visual tracking with online structural similarity-based weighted multiple instance learning. *Inf Sci* 481:292–310
79. NaNa Z, Jin Z (2018) Optimization of face tracking based on KCF and camshift. *Procedia Comput Sci* 131:158–166
80. Chen Z, Huang D, Luo L, Wen M (2020) And, C.Z.: Efficient parallel tld on cpu-gpu platform for real-time tracking. *KSII Trans Int Inf Syst* 14(1):201–220
81. Asha CS, Narasimhadhan AV (2018) Visual tracking using kernelized correlation filter with conditional switching to median flow tracker. *IETE J Res* 66(3):427–438
82. Liu S, Liu D, Srivastava G, Połap D, Woźniak M (2020) Overview and methods of correlation filter algorithms in object tracking. *Complex & Intelligent Systems*
83. Lukežič A, Vojř T, Zajc LČ, Matas J, Kristan M (2018) Discriminative correlation filter tracker with channel and spatial reliability. *Int J Comput Vis* 126(7):671–688
84. ITU-T (1996) Methods for subjective determination of transmissions quality. Recommendation P.800
85. Streijl RC, Winkler S, Hands DS (2016) Mean opinion score (MOS) revisited: methods and applications, limitations and alternatives. *Multimed Syst* 22(2):213–227
86. Dadi H (2020) Quantitative performance metrics for human tracking algorithms. *Int J Anal Experi Modal Anal* 12:7

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Francisco J. Rodríguez-Lozano** received a B.Sc. in Computer Science, M.Sc. in Distributed Renewable Energies, at Ph.D. in Computer Engineering at the University of Córdoba, Spain, in 2016, 2017, and 2020 respectively. He is a member of the Department of Electronics and Computer Engineering at the same university, where he is a full-time Assistant Teacher since 2018. He has been the main author in several papers in high-rank international journals and has

collaborated in different conferences and research national and international projects. He was a member of the Advanced Informatics Research Group (TIC 252) as well as the Lifestyles, Innovation and Health Group (GA16) at the IMIBIC of the University of Córdoba, Spain, since both were created until 2023. His research interests are in the field of image processing, computer vision, machine learning, fuzzy systems, high-performance computing, and robotic systems.



**Juan C. Gámez-Granados** received the MS, Advanced Studies Degree, and Ph.D. degrees, all of them in Computer Science, from the University of Granada, Spain, in 2000, 2006, and 2017 respectively. He has worked in private and public enterprises. Since 2005, he has collaborated with the Department of Electronics and Computer Engineering at the University of Córdoba, Spain, where, since 2007, he works full-time. He has authored papers

in international journals as well as collaborations in books and conferences. These contributions are focused on Artificial Intelligence, image processing, and Fuzzy Logic as well as collaborations with other disciplines. Likewise, He has collaborated on several research projects, as well as competitive programs and R&D contracts with companies. He is a member of the Intelligent Systems Research Group (TIC 175) of the University of Granada as well as the IMIBIC research group (GC20 Genetics and behavioural diseases) of the University of Córdoba.





**Hector Martinez** is a Computer Science Ph.D. at GIIA research group lead by Joaquín Olivares Bueno at University of Cordoba. His predoctoral research was done at the HPC&A group led by Enrique S. Quintana-Orti at Jaume I University. His thesis was focused on the development of a DNA/RNA Aligner and the optimization of different bioinformatic applications with HPC Technics (MPI, OpenMP, Cuda, ...). He had 10 JCR journal papers and more

than 10 International conferences. The last few years, his work has focused on IoT (fog and edge computing), Neuronal Networks and image processing. Now, His interests cover HPC programming models like OpenMP, Cuda, SIMD and MPI High-performance networks and low-power processors.



**Joaquín Olivares** received the B.S. and M.S. degree in Computer Sciences in 1997, and 1999, respectively, and the M.S. degree in Electronics Engineering in 2003, all from the University of Granada, Spain. He received the Ph.D. degree in 2008 at the University of Córdoba, Spain. He is Associate Professor with the Electronic and Computer Engineering Department at the University of Córdoba, Spain, since 2001. He is founder and head of the Advanced Informatics Research Group. His research interests are in the field of the Internet of things, wireless communications, embedded systems, computer vision, and high performance computing.

His research interests are in the field of the Internet of things, wireless communications, embedded systems, computer vision, and high performance computing.



**Jose M. Palomares** was born in Motril, Spain, in 1975. He received his Ph.D., M.Sc., and B.Sc. degrees in Computer Engineering from the University of Granada, Granada, Spain in 2011, 1998, and 1996, respectively. Since 2000, he has been working as a Lecturer, Assistant Professor, and, currently, Associate Professor at the Universidad de Córdoba, Córdoba, Spain. He is co-founder of the Advanced Informatics Research Group (GIIA), Uni-

versidad de Córdoba, Córdoba, Spain. He has research interests in image and video processing, real-time systems, wireless sensor networks, and computer architecture. He is Member of the ACM since 2005.

## Affiliations

Francisco J. Rodríguez-Lozano<sup>1</sup> · Juan C. Gámez-Granados<sup>1</sup> · Héctor Martínez<sup>1</sup> · Jose M. Palomares<sup>1</sup> · Joaquín Olivares<sup>1</sup>

Juan C. Gámez-Granados  
jcgamez@uco.es

Héctor Martínez  
el2mapah@uco.es

Jose M. Palomares  
jmpalomares@uco.es

Joaquín Olivares  
olivares@uco.es

<sup>1</sup> Department of Electronic and Computer Engineering, Universidad de Córdoba, Edificio Leonardo da Vinci, Campus de Rabanales, Córdoba, 14071, Spain