Intelligent Carom Billiards Assistive System for automatic solution path generation and actual path prediction with principal component analysis-based ball motion detection

Jung-kil Park¹ and Jaebyung Park¹,²

Abstract
This article proposes an Intelligent Carom Billiards Assistive System for a novice at Carom Billiards. Since a novice player is very difficult to find the solution paths of the cue ball, the ICBAS automatically calculates the solution paths for given ball configuration which is detected by a monocular vision and displays them in a guidance monitor. For rapid and systematically calculating the solution paths, the five and half system that is the most representative method in the field of billiard is implemented. In addition, the paths of the cue ball changed by the direction of the cue stick are predicted in real time and projected onto the pool table bed. By comparing the solution paths of the cue ball with the projected paths, a player can hit the cue ball easily and successfully. For automatically providing the solution paths and the predicted paths during billiard game, the ball movement is detected by the principal component analysis under linear motion constraints robustly against image noise, occlusion, and so on. Whenever all of the balls stop, both paths are newly calculated and provided for a player. For verifying the feasibility of the proposed Intelligent Carom Billiards Assistive System, the experiments are conducted with the real pool table.

Keywords
Carom billiards, five and half system, monocular camera, ball path generation, principal component analysis

Date received: 11 January 2019; accepted: 24 July 2022

Topic: AI in Robotics; Human Robot/Machine Interaction
Topic Editor: Alessandro Di Nuovo
Associate Editor: Alessandro Di Nuovo

Introduction
According to the report of the Ministry of Culture, Sports and Tourism of the Republic of Korea in 2014, 62.2% of leisure activities are relaxation, but only 8.6% are sports activities. That is because most sports require an understanding of complex rules and a certain level of athletic ability. In addition, in the case of intense sports, the risk of injury can be felt. Unlike other sports, cue sports (or billiard sports) are rarely injured and are popular sports for

¹ Division of Electronics and Information Engineering, Jeonbuk National University, Jeonju-si, South Korea
² Core Research Institute of Intelligent Robots, Jeonbuk National University, Jeonju-si, South Korea

Corresponding author:
Jaebyung Park, Division of Electronics and Information Engineering, Jeonbuk National University, 567 Baekje-daero, Deokjin-gu, Jeonju-si, Jeollabuk-do 54896, South Korea.
Email: jbpark@jbnu.ac.kr
all ages. However, there are still barriers to entry because it takes a lot of time to learn some complicated principles and billiards skills.

There are three major games in cue sports: Carom billiards, Pool (Pocket billiards), and Snooker. The difference between Carom Billiards and Pool and Snooker is that Carom Billiards has no pockets on the pool table whereas Pool and Snooker have pockets. There are many different types of games in Pool and Snooker, but the basic rule is for a player to hit the cue ball to put the object ball into the pocket. Because the rule is relatively simple, many assistive systems have been developed for Pool and Snooker.\(^1\)\(^-\)\(^9\)

Some study was conducted to provide a player with a simple top view of the pool table using intelligent glasses for self-training.\(^1\) However, most of them focus on the cue ball’s path prediction according to the cue stick’s direction and provide the predicted path for a player. The predicted path is obtained by considering collisions with not only cushions but also object balls. In some research studies, physics-based cue ball’s path prediction was studied for a billiards robot, not for a human being. The multiple neural network compensator was developed for a billiard robot to master the cue ball’s trajectory with multiple cushion rebounds,\(^2\) and the fuzzy neural network was proposed to predict the cue ball’s final position after stroking and collision.\(^3\) In another research,\(^4\) the four frame long short-term memory (LSTM) model was proposed for training the predictive dynamics model of billiards with sequential visual inputs. Thanks to the physics-based approach, the cue ball’s trajectory and final position can be obtained more accurately, but it requires higher computational complexity due to neural network training and inference. In previous research studies,\(^5\)\(^-\)\(^6\) a camera was attached to the player’s head and body for ball and cue stick detection. During a billiards match, it may cause inconvenience to the player’s movement. In other research studies, a band and a marker were used for the cue stick detection.\(^7\)\(^-\)\(^8\) In this case, some additional settings are required in advance. Despite these inconveniences, the previous assistive systems seem to be very helpful for a player to find the cue ball path that can put the object ball into the pocket.

Unlike Pool and Snooker, the rule of the Carom billiards is more difficult and complicated. The rule of the three-cushion billiards among games within Carom billiards is that the cue ball hit by a player must hit two object balls and hit the second object ball after contacting at least three cushions. For a novice of the three-cushion billiards, it is challenging to find the solution path of the cue ball satisfying the three-cushion rule. In this article, we propose an Intelligent Carom Billiards Assistive System (ICBAS) for helping a novice player. The proposed ICBAS provides not only the solution paths of the cue ball satisfying the three-cushion rule but also the predicted paths of the cue ball that are changed in real time by the direction of the cue stick. To do so, the ICBAS first detects three balls and a cue stick using a monocular camera with no additional settings required in previous research studies. For robust detection to illumination change, the red–green–blue (RGB) image obtained from the camera is converted into the hue–saturation–value (HSV) image. By applying image preprocessing techniques, such as Gaussian filtering and binary thresholding, and contour extraction technique to the HSV image sequentially, the contour image can be obtained. For detecting three balls, the circle Hough transform (CHT)\(^10\) is applied to the contour image. Finally, each ball color is determined considering the light reflection problem on the balls. Recently, many researchers use convolutional neural networks for edge or contour detection. In the research,\(^11\) the fully convolutional encoder–decoder network is trained end-to-end to effectively highlight object contours with suppressing other edges. In the research,\(^12\) the convolitional oriented boundaries are proposed by considering not only contour strength but also orientation for more accurate results. In the research,\(^13\) the architecture-fusion network is trained end-to-end to effectively reduce the misrecognition of both internal and background edges inside objects without degradation of object detection precision. Deep neural networks (DNNs)-based methods are efficient and useful for object contour detection from various images, but training and inference of DNNs require powerful and expensive systems including graphics processing units (GPUs). Thus, the deep learning system with GPUs seems over-spec to apply to the proposed ICBAS as ball detection on the pool table bed is relatively simple.

With the configuration of the detected three balls, the solution paths of the cue ball can be systematically calculated by using the five and half system\(^14\) that is the most representative method in the field of three-cushion billiards. The ICBAS displays the obtained solution paths in a guidance monitor with the tip-point for the desired spin of the cue ball and the overlap width between the cue ball and the first object ball. A novice player can refer to the solution paths in the monitor during billiard game. However, a player may be difficult to map the solution paths to the actual pool table. Thus, the ICBAS also predicts the cue ball paths changing in real time according to the direction of the cue stick and projects the paths on the actual pool table using a beam projector. That is a kind of augmented reality system. Consequently, a player can successfully play a billiard game conveniently comparing the solution paths and the predicted paths.

Next, it is very important to appropriately provide the solution paths and the predicted paths for a player without interfering with the player’s game. In previous research, it was difficult for the billiard game to proceed smoothly because a player had to trigger using a keyboard, a mouse, and so on manually, to calculate the solution paths every time.\(^15\) To solve this problem, the ICBAS automatically recalculates and provides the solution paths and the predicted paths whenever a player hits the cue ball
and all of the balls stop. In this case, the ICBAS must detect the movement of the balls robustly against false detection caused by image noise, occlusion, and so on. Thus, for detecting the motion of the balls after the player hits the cue ball, the principal component analysis (PCA)\cite{16,17} is used under linear motion constraints since the normal motion of the balls is linear. Of course, the motion of the cue ball may not be linear because of spin, but its motion is eventually linear after the spin disappears. Using the PCA with linear motion constraints, the ICBAS can exactly detect the motion of the balls robustly against the abnormal change of the ball position caused by image noise, occlusion, and so on.

The remaining part of this article is organized as follows. The second section describes the concept of the ICBAS and the flowchart of the proposed algorithm. In the third section, the image processing techniques for detecting three balls and a cue stick are described. In addition, the table bed is detected and the homography transformation from the image coordinates to the object coordinates is obtained from the four corner points of the table bed. In the fourth section, for automatic generation of the solution paths and the predicted paths of the cue ball, the detection method of the ball motion is introduced. In this case, the PCA is used for detecting whether all of the balls stop or not. Next, the solution paths of the cue ball are obtained by using the five and half system in the fifth section. In the sixth section, for verifying the feasibility of the proposed ICBAS, experiments are conducted with an actual pool table. Finally, concluding remarks are described in the seventh section.

**Intelligent Carom Billiards Assistive System**

The basic concept of the proposed ICBAS is shown in Figure 1. The ICBAS provides both the solution paths and the predicted paths for a player to easily make a successful billiard shot. The solution paths are calculated by using the five and half system that is the most representative technique in the field of billiards to systematically find the paths satisfying the three-cushion rule. The resultant solution paths are displayed together with the tip-point for the required spin of the cue ball and the overlap width between the cue ball and the first object ball in a guidance monitor. For player’s convenience, the guidance monitor is continuously controlled to head for a player during billiard game. For automatic generation of the solution paths, the three ball positions are detected by a monocular camera. Next, the actual paths of the cue ball are predicted and projected by a beam projector onto the pool table bed. According to the change of the cue stick direction, the predicted paths are repeatedly recalculated and reprojected as shown in Figure 1. In this case, a player changes the direction of the cue stick to find the predicted path that is coincided with the solution path in the guidance monitor and then hits the cue ball along the predicted path. In this way, the proposed ICBAS helps a player enjoy his or her own game even though he or she is a beginner of Carom Billiards.

In Figure 2, the algorithm of the ICBAS is depicted as the flowchart in detail. The image obtained from the camera is first calibrated by using the GML C++ Calibration...
Next, the calibrated image is preprocessed for noise reduction and contour detection. From the detected contour image, the pool table bed and the three balls are detected. Using the obtained corner points of the table bed, the homography transformation matrix is calculated from the image coordinates to the object coordinates. Using the homography transformation matrix, the three ball positions in the object coordinates can be calculated from the points detected in the image coordinates. Next, the ICBAS detects the motion of the balls by applying the PCA to the trajectory points of the balls represented in the object coordinates. If all of the balls stop, the ICBAS automatically finds the solution paths of the cue ball and displays them in the guidance monitor. In addition, the cue ball path according to the direction of the cue stick is predicted and projected onto the table bed. If at least one of the balls begins to move, the ICBAS just receives the next image from the camera and continuously checks the ball motion until all of the balls stop again. In this way, the ICBAS appropriately provides the solution paths and the predicted paths for a player during billiard game. The detail methods are described in the following sections.

**Ball and cue stick detection**

In this section, the detection method of three balls and a cue stick is introduced based on the image processing techniques. For robust detection to illumination change and image noise, the ICBAS first detects the pool table bed from the contour image and sets the region of interest (ROI) as the quadrilateral of the table bed. Within the ROI, three ball positions are obtained by the circle Hough transform (CHT). Next, the color of each ball is detected by averaging color robustly against the light reflection on the surface of the ball. For detecting the direction of the cue stick, the ICBAS first sets the ROI around the cue ball and detects the center position of the cue stick within the ROI. Finally, the direction of the cue stick is determined as the vector from the center of the cue stick to the center of the cue ball. The detail process is described as follows.

The image from the monocular camera is preprocessed as shown in Figure 3. First, the original RGB image is converted into the HSV image, where V, S, and H values of the HSV color model are calculated from R, G, and B values of the RGB color model as in equations (1), (2) and (3), respectively.

\[
V = \max(R, G, B) \quad (1)
\]

\[
S = \begin{cases} 
\frac{(V - \min(R, G, B))}{V} & \text{if } V \neq 0 \\
0 & \text{if } V = 0 
\end{cases} \quad (2)
\]

\[
H = \begin{cases} 
\frac{(G - B) \times 60}{S} & \text{if } V = R \\
120 + \frac{(B - R) \times 60}{S} & \text{if } V = G \\
240 + \frac{(R - G) \times 60}{S} & \text{if } V = B 
\end{cases} \quad (3)
\]

where \( H \) is set as \((H + 360)\) for \( H < 0 \). In this case, \( 0 \leq H \leq 179 \), \( 0 \leq S \leq 255 \), and \( 0 \leq V \leq 255 \). Next, the HSV image is thresholded for detecting the green table bed as in equation (4).

\[
(T^H, T^K, T^V) \leq (H, S, V) \leq (T^U_H, T^U_S, T^U_V) \quad (4)
\]

where \( T^K \) and \( T^K_u \) for \( K = \{H, S, V\} \) are lower and upper threshold values, respectively. For detecting the green table bed, the actual threshold values of \((T^H, T^K, T^V)\) and \((T^U_H, T^U_S, T^U_V)\) are determined as \((80, 100, 10)\) and \((100, 255, 200)\), respectively. The hue value \( H \) is determined within the table green color range with a small offset value.
However, the lower threshold of the saturation $S$ is less than half and the upper threshold is the maximum, so the range of the saturation $S$ is very wide. Furthermore, the range of the brightness $V$ covers almost the entire range. As a result, it is possible to detect the pool table without being greatly affected by illumination change. From the binary image, the contours are detected by the border following technique\textsuperscript{19} as shown in Figure 3.

For detecting the pool table bed, the contours are first approximated as polygons by using the Douglas–Peucker approximation.\textsuperscript{20} Since the largest quadrilateral indicates the table bed as shown in the contour extraction image of Figure 3, detecting the table bed is like finding the largest quadrilateral among the polygons. The quadrilaterals with four corner points are extracted among the polygons and their sizes are compared to find the largest quadrilateral. In this case, we assume the size of the table bed in the camera image is larger than $\frac{1}{4}$ of the whole image size and thus set the initial size of the quadrilateral as $\frac{W_{\text{img}}}{2} \times \frac{H_{\text{img}}}{2}$, where $W_{\text{img}}$ and $H_{\text{img}}$ are the width and height of the image. In this way, the four corner points of the table bed in the image coordinates of $\{I\}$ can be obtained as $^{I}P_1, ^{I}P_2, ^{I}P_3,$ and $^{I}P_4$ in Figure 4, where the superscripts indicate the coordinates. The obtained corner points in $\{I\}$ and the physical corner points, $^{O}P_1, ^{O}P_2, ^{O}P_3,$ and $^{O}P_4$, in the object coordinates of $\{O\}$ are related each other by homography between them as in equation (5).

$$s_i^{O}P_i = \mathbf{H}^{I}P_i \quad \text{for} \quad i = 1, 2, 3, 4$$ (5)
where \( \mathbf{H} \) is the 3 \( \times \) 3 homography transformation matrix, \( s_i \) is the scale factor, and \( P \) and \( O \) for \( i = 1, 2, 3, 4 \) are represented as \([x_i', y_i']^T\) and \([x_i, y_i]^T\) in homogeneous coordinates, respectively. The homography transformation matrix \( \mathbf{H} \) is obtained by minimizing the back-projection error of \( E \) in equation (6):21

\[
E = \sum_i \left( \left( \frac{O_{x_i} - h_{11}x_i + h_{12}y_i + h_{13}}{h_{31}x_i + h_{32}y_i + h_{33}} \right)^2 + \left( \frac{O_{y_i} - h_{21}x_i + h_{22}y_i + h_{23}}{h_{31}x_i + h_{32}y_i + h_{33}} \right)^2 \right)
\]  

(6)

where \( h_{ij} \) for \( i = 1, 2, 3 \) and \( j = 1, 2, 3 \) are elements of \( \mathbf{H} \). In this case, the denominator, \((h_{31}x_i + h_{32}y_i + h_{33})\), is the scale factor of \( s_i \) in equation (5). The corresponding physical points are set as the corner points of the actual pool table without wooden rails as in equations (7)–(10).

\[
O_P_1 = \begin{bmatrix} \frac{W_T}{W_W} & W_W \\ W_T - W_W & 1 \end{bmatrix}
\]

(7)

\[
O_P_2 = \begin{bmatrix} W_W \\ W_T - W_W \\ 1 \end{bmatrix}
\]

(8)

\[
O_P_3 = \begin{bmatrix} \frac{L_T - W_T}{W_W} \\ W_T - W_W \\ 1 \end{bmatrix}
\]

(9)

\[
O_P_4 = \begin{bmatrix} \frac{L_T - W_T}{W_W} \\ W_W \\ 1 \end{bmatrix}
\]

(10)

where \( W_T \) and \( L_T \) are the width and length of the pool table, and \( W_W \) is the width of the wooden rail. Using the obtained homography transformation matrix \( \mathbf{H} \), any point \( iP \) in \( \{I\} \) can be transformed to point \( O \) \( iP \) in \( \{O\} \) as in equation (11).

\[
sO_P = \mathbf{H}iP
\]

(11)

where \( s \) is the scale factor.

Next, we detect three balls from the contour image of Figure 3 as shown in Figure 5. We first apply the Gaussian smoothing filter to the contour image for reducing noise and small details, where the two-dimensional (2D) Gaussian filter is defined in equation (12).

\[
G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2 + y^2}{2\sigma^2}}
\]

(12)

where \( \sigma \) is the standard deviation of the Gaussian distribution, and the mean is \((0, 0)\). Using equation (12), the Gaussian kernel is computed as an \( n \times n \) matrix, where \( \sigma \) is set as \((\frac{r}{2} - 1) \times 0.3 + 0.8\). By convolving the contour image with the Gaussian kernel, the smoothed image can be obtained as in the first image of Figure 5. Next, we find circles from the smoothed image by using the circle Hough transform (CHT).10 The CHT is basically based on the voting procedure like the original Hough transform (HT).22,23 However, the CHT differs from the HT in that the parameter space of the CHT is 3D whereas that of the HT is 2D. Mathematically, the equation of a circle in \( x-y \) plane is given by equation (13).

\[
(x - a)^2 + (y - b)^2 = r^2
\]

(13)

where \((a, b)\) is the center of the circle, and \( r \) is the radius. For point \((x, y)\) in the image, the CHT finds all possible parameters, \( a, b, \) and \( r \), according to equation (13), and votes each grid cell \((a, b, r)\) in the 3D parameter space \( a-b-r \) (or Hough space), where voting means that the voting number of the cell increases by 1. Among the voted grid cells in the \( a-b-r \) space, the grid cell whose voting number is the local maxima is selected to determine the resultant circle parameters. In this case, we define the minimum distance \( D_{\text{min}} \) among circles to remove the falsely detected circles around the true one. In other words, if the distances among the grid cells with the local maxima are less than \( D_{\text{min}} \), all of the grid cells are removed except the one grid...
cell with the largest voting number. We also define the minimum radius $R_{\text{min}}$ and the maximum radius $R_{\text{max}}$ of the circle, taking the actual radius of the ball into consideration. In other words, the radius of the detected circle should be in the range from $R_{\text{min}}$ to $R_{\text{max}}$. As a result, the detected circles for three balls are shown in the second image of Figure 5, where $I^P_{P1}, I^P_{P2}$, and $I^P_{P3}$ are their center positions in $\{I\}$. Next, the colors of the balls are determined by averaging the colors of the $(N + 1)^2$ points around the center position $(x_c, y_c)$ of the ball as in equation (14) for robust color detection to light reflection as shown in Figure 6.

$$C(x_c, y_c) = (\bar{R}, \bar{G}, \bar{B})$$

where

$$\bar{R} = \frac{1}{(N + 1)^2} \sum_{j=0}^{N} \sum_{i=0}^{N} R(X_i, Y_j)$$

$$\bar{G} = \frac{1}{(N + 1)^2} \sum_{j=0}^{N} \sum_{i=0}^{N} G(X_i, Y_j)$$

$$\bar{B} = \frac{1}{(N + 1)^2} \sum_{j=0}^{N} \sum_{i=0}^{N} B(X_i, Y_j)$$

where $R(X_i, Y_j)$, $G(X_i, Y_j)$, and $B(X_i, Y_j)$ are red, green, and blue colors at $(X_i, Y_j)$ position, respectively, and $(X_i, Y_j)$ is determined as $(x_c - W_c + i \times \Delta d, y_c - W_c + j \times \Delta d)$. In this case, $N$ is $\lceil \frac{W}{3} \rceil$, where $\lceil \cdot \rceil$ discards the decimal places. If $\bar{R}, \bar{G},$ and $\bar{B}$ at $(x_c, y_c)$ satisfy the condition in equation (18), the circle with the center position $(x_c, y_c)$ is determined as the red ball.

$$\bar{R} \geq \bar{G} \times 1.5 \text{ and } \bar{R} \geq \bar{B} \times 1.5$$

If the condition in equation (18) is not satisfied but the condition in equation (19) is satisfied, the circle is determined as the yellow ball.

If both conditions in equations (18) and (19) are not satisfied, the circle is determined as the white ball. In this way, we can determine the positions of the red, yellow, and white balls, $I^P_{R}, I^P_{Y},$ and $I^P_{W}$, as shown in the third image of Figure 5.

Next, the three ball positions in the image coordinates of $\{I\}$ are transformed to the physical positions in the object coordinates of $\{O\}$ by the homography transformation matrix $H$ in equation (5) as follows.

$$O^P_{PK} = H^I P_K \text{ for } K = W, Y, R$$

The resultant ball positions, $O^P_{W}, O^P_{Y},$ and $O^P_{R}$, in $\{O\}$ are depicted in Figure 7 and will be used as the ball positions in the following sections.

Next, the cue stick is detected as shown in Figure 8. For detecting the cue stick, the HSV image of Figure 8(a) is thresholded by the thresholding values in equation (21).

$$20, 20, 0 \leq (H, S, V) \leq 20, 80, 255$$

Similar to the table detection, the binary image of the cue stick can be extracted without being greatly affected by illumination change using the HSV color model. The Gaussian smoothing filter with the size of $3 \times 3$ and the operations of erosion and dilation are sequentially applied to the thresholded image for noise reduction. Next, we define the ROI with the width of $W_S$ at the cue ball position because the cue stick detection is needed only for predicting the path of the cue ball, where the cue ball is the white ball. In this case, the square with the width of $W_S$ is subtracted from the ROI to remove the falsely detected points of the cue ball because the color of the cue stick is almost same as the white ball. The center position of the object points in the ROI is obtained as the center position $O^P_S$ of the cue stick. Using the homography transform in equation (11), we can obtain the cue center position of $O^P_S$ and the cue ball position of $O^P_{W}$ in the object coordinates of $\{O\}$, where $O^P_{W}$ is obtained previously as in
Figure 7. Eventually, we can find the direction of the cue stick as the vector $\overrightarrow{PS} \overrightarrow{PW}$. Similar to the white cue ball, the direction of the cue stick can be obtained for the yellow cue ball as $\overrightarrow{PS} \overrightarrow{PY}$. According to the direction of the cue stick, we can predict the cue ball path under the conditions such as perfectly elastic collision between balls and the same angle of incidence and reflection of the ball.

PCA-based ball motion detection

For appropriately providing the solution paths and the predicted paths without interfering player’s game, the new paths must be automatically recalculated and provided for a player whenever the player hits the cue ball and then all of the balls stop. In this section, the detection method for the moment when a player hits the cue ball and balls are moving and when all of the balls stop is proposed. In this case, we assume that the normal motion of the balls is linear. Of course, the motion of the cue ball may not be linear at the moment when a player hits the ball with spin, but the motion of the ball eventually becomes linear after the spin disappears. Thus, we define the linear motion constraints to detect the ball motion robustly against falsely detected ball positions. The false detection of the ball positions may occur due to image noise or occlusion by the player preparing the next billiard shot. Thus, the PCA is applied to the trajectory of the ball to robustly determine whether the ball motion satisfies the linear motion constraints. If the ball trajectory satisfies the linear motion constraints, it is determined that the ball has started to move. Next, it is determined that the ball has stopped if the variances of the ball positions are small enough. In this case, the variances of the ball positions can be obtained by the moment when a player hits the cue ball and balls are moving and when all of the balls stop is proposed. In this section, the detection method for the moment when a player hits the cue ball and balls are moving and when all of the balls stop is proposed. In this case, we assume that the normal motion of the balls is linear. Of course, the motion of the cue ball may not be linear at the moment when a player hits the ball with spin, but the motion of the ball eventually becomes linear after the spin disappears. Thus, we define the linear motion constraints to detect the ball motion robustly against falsely detected ball positions. The false detection of the ball positions may occur due to image noise or occlusion by the player preparing the next billiard shot. Thus, the PCA is applied to the trajectory of the ball to robustly determine whether the ball motion satisfies the linear motion constraints. If the ball trajectory satisfies the linear motion constraints, it is determined that the ball has started to move. Next, it is determined that the ball has stopped if the variances of the ball positions are small enough. In this case, the variances of the ball positions can be obtained by the principal components. The detail method is described as follows.

For detecting the moment when a ball moves or stops, we apply the PCA to the $n$ latest ball positions, $p(t), p(t-1), ..., p(t-(n-2)), p(t-(n-1))$, in the object coordinates of $\{O\}$. For convenience of explanation, let the ball positions $p(t), ..., p(t-(n-1))$ be $(x_1, y_1), ..., (x_n, y_n)$, respectively. The ball positions can be represented as an $n \times 2$ matrix $X$ in equation (22).

$$X = \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ \vdots \\ x_n & y_n \end{bmatrix}$$

(22)

where the mean $\bar{X}$ of $\{x_1, x_2, ..., x_n\}$ and the mean $\bar{Y}$ of $\{y_1, y_2, ..., y_n\}$ are defined in equations (23) and (24), respectively.

$$\bar{X} = E(X) = \frac{1}{n} \sum_{i=1}^{n} x_i$$

(23)

$$\bar{Y} = E(Y) = \frac{1}{n} \sum_{i=1}^{n} y_i$$

(24)

With these means, the center position of the $n$ ball positions is determined as $P_c(\bar{X}, \bar{Y})$. Next, the covariance between two dimensions for $X$ and $Y$ can be obtained based on Bessel’s correction as in equation (25).

$$\sigma_{xy} = \text{cov}(X, Y) = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{X})(y_i - \bar{Y})$$

(25)

where $\sigma_{xy}$ is equal to $\sigma_{yx}$ because the covariance satisfies the commutative law. The covariance between one dimension and itself is the variance. For each dimension, the variance is obtained as in equations (26) and (27).

$$\sigma_{xx} = \text{cov}(X, X) = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{X})(x_i - \bar{X})$$

(26)

$$\sigma_{yy} = \text{cov}(Y, Y) = \frac{1}{n-1} \sum_{i=1}^{n} (y_i - \bar{Y})(y_i - \bar{Y})$$

(27)
Figure 9. PCA application to \( n \) consecutive ball positions from \((x_1, y_1)\) to \((x_n, y_n)\), where \( P_c(t) \) is a center position at time \( t \), and \( V_1 \) and \( V_2 \) are eigenvectors for eigenvalues \( \lambda_1 \) and \( \lambda_2 \) respectively (\( \lambda_1 \geq \lambda_2 \)).

From equation (25) to equation (27), the covariance matrix \( C \) can be defined in equation (28).

\[
C = \begin{bmatrix} \sigma_{xx} & \sigma_{xy} \\ \sigma_{yx} & \sigma_{yy} \end{bmatrix}
\]  

(28)

where \( C \) is a \( 2 \times 2 \) symmetric matrix, and its eigenvalue \( \lambda \) and the eigenvector \( V \) can be obtained from equation (29).

\[
CV = \lambda V
\]  

(29)

where two eigenvectors, \( V_1 \) and \( V_2 \), are the principal components of \( X \) and are mutually perpendicular (or orthogonal) as shown in Figure 9. The eigenvalues \( \lambda_1 \) and \( \lambda_2 \) are variances in the directions of \( V_1 \) and \( V_2 \) respectively, where \( \lambda_1 \geq \lambda_2 \). In Figure 9, the center position \( P_c(t) \) at time \( t \) is determined as \((X, Y)\) from equations (23) and (24).

First, we define the condition for ball stop detection with the largest variance \( \lambda_1 \) as in equation (30) when the ball has been moving.

\[
\sqrt{\lambda_1} < D_S
\]  

(30)

where \( \sqrt{\lambda_1} \) is the largest standard deviation of \( X \), and \( D_S \) is defined by considering the maximum position error caused by image noise as shown in Figure 10. The fact that all of three balls satisfy the condition in equation (30) means that all of the balls have stopped. At this time, their center positions are prepared for the next steps such as finding the solution paths and predicting the real-time path of the cue ball. In this case, the obtained center positions for the next step are no longer updated because slight change of the ball positions may be caused by image noise or occlusion by a player preparing the next billiard shot. The next update will be performed when a player hits the cue ball and all of the balls stop again.

Next, as shown in Figure 11, we define the conditions for ball motion detection caused by player’s hitting as in equations (31) and (32) when all of the balls have stopped according to equation (30). First, for removing the abnormal ball motion change caused by falsely detected ball positions and detecting true ball motion by actual player’s hitting, we detect only linear motion of the ball by using the condition in equation (31).

\[
\sqrt{\lambda_1} > D_{M1} \text{ and } \sqrt{\lambda_2} < D_{M2}
\]  

(31)

where \( D_{M1} \) is the minimum displacement for true ball motion detection, and \( D_{M2} \) is the admissible displacement error in the perpendicular direction to ball motion. For linear ball motion, \( D_{M2} \) is defined much smaller than \( D_{M1} \) and near zero. Next, we define the condition for ball moving at almost constant speed as in equation (32).

\[
\frac{D_{CH}}{D_{CT}} < 1.1
\]  

(32)

where \( D_{CH} \) is the distance between the center position \( P_c(t) \) and the position of the heading point \( P(t) \), and \( D_{CT} \) is the distance between the center position \( P_c(t) \) and the position of the tail point \( P(t - (n - 1)) \). In other words, the condition in equation (32) means that the moving distances of the ball before and after the center position are almost the same. In fact, \( D_{CH} \) is smaller than \( D_{CT} \) and the left side of equation (32) is less than 1 because the speed of the ball decreases in time. However, we determine the right side of equation (32) as 1.1 instead of 1 considering the detection error of the ball positions. If the abrupt change of the heading point is occurred by image noise or occlusion, \( X \) cannot satisfy the conditions in equation (31) or equation (32). In this way, the false detection for the ball moving can be prevented. When at least one of three balls satisfies both conditions in equations (31) and (32), it is determined that the ball motion occurs.

**Systematic solution path acquisition of the cue ball**

The basic rule of the three-cushion billiards among games within Carom billiards is that the cue ball hit by a player must hit two object balls and hit the second object ball after contacting at least three cushions. For finding the solution
path of the cue ball that satisfies the rule, we use the five and half system (or the diamond system)\(^{14}\) that is the most representative technique for Carom billiards as shown in Figure 12, where the spin amount is basically set to the maximum at the middle height of the cue ball. If a player hit the cue ball, the cue ball begins to move from the starting point of \(S\) to the aiming point of \(A\). After contacting the first cushion on the long rail, the cue ball contacts the second cushion on the short rail. After contacting the second cushion, the cue ball moves to the finishing point of \(F\) on the third cushion. After contacting the third cushion, the cue ball finally moves to the object balls and hits the balls. The five and half system defines the relationship among these three points of \(S\), \(A\), and \(F\) by the following formula in equation (33).

\[
A = S - F
\]  

For the given starting point \(S\) and finishing point \(F\), the aiming point of \(A\) can be calculated by equation (33), where \(S\), \(A\), and \(F\) are scalar.

In the five and half system, for calculating the formula in equation (33), the scalar values of the marker points on the wooden rail are defined nonlinearly and differently for \(S\), \(A\), and \(F\) as shown in Figure 13. In this case, \((S - F)\) must be greater than or equal to zero because \(A\) is greater than or equal to zero.

For finding the solution path of the cue ball, we first find the three cushion line to hit the object balls. For given finishing points, three cushion lines are determined as shown in Figure 14. In this case, we can find the three cushion line of the finishing point \(F\) of 20 to hit the object balls. The three cushion line between each two lines is calculated by linear interpolation.

Next, for the given position of the cue ball, we find the starting point \(S\) and the aiming point \(A\) that satisfy the formula in equation (33) as shown in Figure 15. For the given finishing point \(F\) of 20, the starting point \(S\) and the aiming point \(A\) are determined by equation (33) as 40 and 20, respectively. In this way, we can find the solution path of the cue ball. During billiard game, a player just hits the cue ball for the aiming point \(A\) and then the cue ball moves along the solution path to hit the object balls.

So far we find the solution path of the cue ball in the case of two object balls attached. In the case of two object balls separated, the solution path can be found as shown in Figure 16. In this case, we find the solution path for the cue ball to hit the first object ball and then to hit the second object ball after contacting three cushions. To do
so, we first find the three cushion line to hit the second object ball. Next, we define the ghost ball right next to the first object ball by considering the collision point between the cue ball and the first object ball. In this case, we regard the ghost ball as the cue ball and calculate the starting point $S$ and the aiming point $A$ similar to the attached object ball case.

Finally, we determine the direction of the cue ball so that its tangent after collision with the first object ball points to the aiming point $A$ of 20. If we assume that the collision between balls is perfectly elastic, the collision can be described as shown in Figure 17. The aiming velocity $v_a$ of the cue ball is divided into the velocity $v_o$ of the first object ball and the velocity $v_c$ of the cue ball after the collision. The direction of $v_a$ is determined parallel with the vector $P_{CB} - P_{OB}$ where $P_{CB}$ and $P_{OB}$ are the center points of the cue ball and the object ball, respectively. The vector $v_c$ on the tangent line is determined perpendicular to $v_o$. Using the angle $\theta_o$ between $v_a$ and $v_o$, the angle $\theta_c$ between $v_a$ and $v_c$ can be calculated by equation (34).

\[
\theta_c = 90^\circ - \theta_o \tag{34}
\]

When the vector $v_c$ is coincided with the tangent line, the overlap width $d_t$ between the two balls in the direction of $v_a$ can be calculated by geometry. According to the overlap width corresponding to $\theta_o$ and $\theta_c$, the fractional aiming table can be defined as shown in Figure 18 based on the fractional aiming system. During billiard game, if a player just hits the cue ball with the given overlap width, the cue ball moves along the solution path after hitting the first object ball and finally hits the second object ball.

When applying the five and half system, we can find up to two solution paths depending on which ball is set as the first object ball. In addition, for each case, we can find up to four solution paths by symmetrically transforming the ball positions vertically or horizontally. Furthermore, we can find more paths by anticipating the path after contacting the fourth cushion, where the cue ball after the fourth cushion simply moves at the same incidence and reflection angles since the amount of the spin of the cue ball is almost gone. In this way, the solution paths can be found and displayed in the guidance monitor. Although we use only the five and half system for finding the solutions paths in this article, much more solution paths can be...
found by various kinds of billiard techniques such as the plus system, the maximum English system, the double rail system, the reverse English system, and the no English system.

**Experimental results**

**Experimental setup**

For verifying the feasibility of the proposed ICBAS, experiments are conducted in a 60% reduced pool table as shown in Figure 19. For detecting the balls and the cue stick, a monocular camera, Logitech C615 webcam is installed 1.78 m above the pool table at a 27° from the ceiling with a beam projector. Using the camera detection results, the control PC calculates the predicted path and projects it onto the pool table bed by the beam projector. Also, the PC calculates the solution paths for three-cushion billiards and displays them in the guidance monitor which has a direction controller for the monitor to head for a player. The direction controller is implemented using a dynamixel AX-12+ servo motor with an Arduino nano. The Arduino receives commands to control the servo motor from the PC by serial communication. The detail parameters for experiments are described in Table 1.

**Results of automatic solution path providing**

The ICBAS calculates the solution paths of the cue ball and displays the paths in the guidance monitor. The trajectories of three balls during billiard game are shown in Figure 20, where the trajectories of the white, yellow, and red balls are depicted as blue, red, and green points, respectively. In Figure 20(a), all of the balls stop before the player hits the white ball as the cue ball. In Figure 20(b), as soon as the player hits the white ball, the ball moves too fast to get several positions of the ball. In Figure 20(c), the red ball begins to move after colliding with the white ball. In this case, the red ball moves as fast as the white ball, so it gets almost no position for the red ball. However, from Figure 20(d), the positions of the white and red balls are

---

**Table 1. Parameters for experiments.**

| Parameters | Descriptions |
|------------|--------------|
| $W_{img} = 1920$ pixels | Width of the camera image |
| $H_{img} = 1080$ pixels | Height of the camera image |
| $W_T = 880$ mm | Width of the pool table |
| $L_T = 1600$ mm | Length of the pool table |
| $W_{WR} = 30$ mm | Width of the wooden rail |
| $R_B = 29$ mm | Radius of the ball |
| $n = 11$ | Gaussian kernel matrix size of $(n \times n)$ for table bed detection |
| $D_{min} = 35$ pixels | Minimum distance to remove the falsely detected circles for CHT |
| $R_{min} = 15$ pixels | Minimum radius of the detected circle for CHT |
| $R_{max} = 50$ pixels | Maximum radius of the detected circle for CHT |
| $W_S = 240$ pixels | Width of ROI for cue stick detection |
| $W_B = 80$ pixels | Width of rejection square for cue stick detection |
| $N_S = 10$ | Number of ball positions for PCA-based ball motion detection |
| $D_S = 3$ mm | Admissible position error for ball stop detection |
| $D_{M1} = 30$ mm | Minimum displacement for ball motion detection |
| $D_{M2} = 1.5$ mm | Admissible displacement error in the perpendicular direction to the ball motion |

ROI: region of interest; PCA: principal component analysis.
Figure 20. Three ball trajectories during billiard game where the trajectories of the white, yellow, and red balls are depicted as blue, red, and green points, respectively: (a) All of the balls stop before player’s hitting. (b) The white ball is moving too fast at the moment of player’s hitting. (c) After colliding with the white ball, the red ball moves quickly, either. (d) The movement of both white and red balls can be confirmed through their trajectories. (e) Whole trajectories of all three balls can be confirmed until all of the balls stop after the player hits the white ball as shown in (b).
sufficiently obtained to confirm their linear motion. In Figure 20(e), the whole trajectories of the three balls are shown until all of the balls stop after player’s hitting.

During billiard game in Figure 20, the ball motion is detected as shown in Figure 21. In this case, only NS latest positions for each ball are depicted where NS is 10 as in Table 1, and the circle indicates the center position \( P_c(t) \). The black solid line from the center position indicates the vector \( V_1 \) of the first principal axis with variance \( l_1 \), and the gray line indicates the vector \( V_2 \) of the second principal axis with variance \( l_2 \) although it is sometimes short and thus not well visible. Before all of the balls stop as in Figure 20(a), only the yellow ball moves whereas the other balls stop, and the trajectories of the balls are shown in Figure 21(a). In Figure 21(b), three balls are judged to be stopped because all of them satisfy the ball stop condition in equation (30). That is, if the variances of the three ball positions are small enough, we judge that there is no motion of the balls or all of the balls stop. After the player hits the white ball as shown in Figure 20(b), the white ball begins to move, and the red ball also begins to move due to the collision with the white ball. However, in this case, the motion of the white ball does not satisfy the linear ball motion conditions because the condition of \( \sqrt{l_2} < D_{M2} \) is not satisfied, where \( \sqrt{l_2} \) and \( D_{M2} \) are 77.27 and 1.5, respectively. In Figure 21(d), both linear ball motion conditions of \( \sqrt{l_1} > D_{M1} \) and \( \sqrt{l_2} < D_{M2} \) in equations (31) and (32) are satisfied, where \( \sqrt{l_1}, \sqrt{l_2}, \) and \( D_{M1} \) are 86.55, 1.36, and 30, respectively. At this time, the ICBAS eventually judges that the motion of the balls occurs. Otherwise, the ICBAS may incorrectly judge the motion of the ball by falsely detected positions of the balls due to image noise or occlusion by a player preparing the next shot. After the balls move like in Figure 20(e), all of the balls are judged to be stopped again.

**Figure 21.** Ball motion detection results: (a) Yellow ball is moving with \( \sqrt{l_1} = 6.44 \) and \( \sqrt{l_2} = 1.93 \) at \( t = 11 \). (b) Yellow ball stops with \( \sqrt{l_1} = 2.94 \) and \( \sqrt{l_2} = 1.42 \) at \( t = 41 \). (c) White ball is moving after player’s hitting with \( \sqrt{l_1} = 130.86 \) and \( \sqrt{l_2} = 77.27 \) at \( t = 241 \) but the linear ball motion constraints are not satisfied because of the large value of \( \sqrt{l_2} \). (d) White ball is moving with \( \sqrt{l_1} = 86.55 \) and \( \sqrt{l_2} = 1.36 \) at \( t = 256 \) and the linear ball motion constraints are satisfied. (e) All of the balls stop with \( \sqrt{l_1} = 2.97 \) and \( \sqrt{l_2} = 1.01 \) for white ball, \( \sqrt{l_1} = 1.33 \) and \( \sqrt{l_2} = 0.83 \) for yellow ball, and \( \sqrt{l_1} = 1.45 \) and \( \sqrt{l_2} = 0.83 \) for red ball at \( t = 453 \).
because the trajectories of three balls satisfy the ball stop condition in equation (30) as shown in Figure 21(e). In this way, the ICBAS can detect the moment when all of the balls stop or when the motion of the balls occurs.

Figure 22 shows the case that the white ball is falsely detected due to the occlusion by a player preparing the next billiard shot. (a) All of the balls stop and their trajectories satisfy the ball stop condition in equation (30). (b) The white ball is occluded by the arm of the player preparing the next billiard shot and the abrupt change of the white ball position occurs. Due to the abrupt change of the ball position, the constant ball speed condition in equation (32) cannot be satisfied. (c) The white ball is still occluded and there are several positions falsely detected. In this case, the linear ball motion condition in equation (31) is not satisfied because the gray line represents the value of $\frac{1}{\sqrt{2}}$ and is larger than $D_{M2}$ of 1.5 mm. (d) The occluded white ball appears again and the variance of its positions becomes small enough again.

Figure 22 shows the case that the white ball is falsely detected due to the occlusion by a player preparing the next billiard shot. In Figure 22(a), all of three balls stop and the motion of three balls satisfy the ball stop condition in
equation (30) since their variances are small enough. In Figure 22(b), the white ball begins to be occluded by the arm of the player and the abrupt change of the white ball position occurs because the ICBAS incorrectly recognizes the player’s arm as the white ball. In this case, the trajectory of the white ball cannot satisfy the constant speed condition in equation (32) due to the abrupt change of the ball position. Thus, the ICBAS does not judge that the motion of the balls occurs. In Figure 22(c), several falsely detected positions are shown due to the motion of the player. In this case, the trajectory of the white ball cannot satisfy the linear motion condition of $l^2 < \frac{p}{2} < D_{M2}$ in equation (31), where the gray line from the center position represents the value of $\sqrt{l^2}$ and is larger than $D_{M2}$ of 1.5 mm in Table 1. Thus, the ICBAS maintains the judgment that all of the balls stop. In this way, the ICBAS can prevent from judging incorrectly the ball motion due to the occlusion. In Figure 22(d), the white ball appears and thus its variance becomes small enough again.

Whenever all of the balls are judged to be stopped, the solution paths are calculated and displayed in the guidance monitor. In Figure 23, the result of two solution paths for the white cue ball is depicted. In the first case, the yellow ball is the first object ball and the red ball is the second object ball. By using the five and half system, the solution path can be found and displayed together with the tip-point for maximum spin and the overlap width between the cue ball and the first object ball. In the second case, the red ball is the first object ball and the yellow ball is the second object ball. In this case, the path after contacting the fourth cushion is first estimated to hit the second object ball, where the path after the fourth cushion satisfies the condition of the same angle of incidence and reflection of the ball because the spin of the cue ball almost gone. For the fourth cushion of the estimated path, the solution path is found by the five and half system. The resultant solution path is also displayed with the tip-point and the overlap width.

In Figure 24, two solution paths for the yellow cue ball are depicted. In the first case, the white ball is the first object ball and the red ball is the second object ball. The solution path can be found by using the five and half system and displayed with the tip-point and the overlap width like the first case in Figure 23. In the second case, although the white ball is the first object ball and the red ball is the second object ball as in the first case, we estimate the path after contacting the fourth cushion and find the solution path for the fourth cushion like the second case in Figure 23. The resultant solution path is also displayed with the tip-point and the overlap width. During billiard game, a player can refer to the solution paths with the tip-point for the desired ball spin and the overlap width displayed in the guidance monitor.

**Results of real-time path projection**

The ICBAS detects the three ball positions and the cue stick direction and projects the cue ball path according to the ball configuration and the cue stick direction as shown in Figure 25. In Figure 25(a), only the predicted path of the cue
ball is projected because of no collision with the object balls before the third cushion. On the other hand, in Figure 25(b), both paths of the cue ball and the object ball are projected together because the cue ball collides with the object ball before the third cushion, where the collision occurs between the first cushion and the second cushion. Also, it can be confirmed that the direction of the guidance monitor displaying the solution paths of the cue ball is continuously controlled by detecting the cue stick position to head for the player during player’s preparing the next billiard shot.

The actual motion of the balls after player’s hitting is shown in Figure 26. In Figure 26(a), the predicted paths of the cue ball and the object ball are shown before player’s hitting. In Figure 26(b), it can be confirmed that the actual movement of the balls from 1 to 9 are almost same with the predicted path.
Figure 26. Actual motion of the balls after player’s hitting: (a) predicted cue ball path and (b) actual movement from 1 to 9 after player’s hitting.
Conclusions
In this article, the ICBAS has been proposed to provide both the solution paths and the predicted paths of the cue ball for a novice player of Carom Billiards to easily enjoy his or her own game. Whenever the ICBAS detects the moment when the player hits the cue ball and all of the balls stop, the ICBAS automatically recalculates both the solution paths and the predicted paths of the cue ball. In this case, the motion of the balls is robustly detected by the PCA under the linear ball constraints against the false detection for the ball positions due to image noise or occlusion by the player preparing billiard shot. The solution paths are systematically calculated by using the five and half system and displayed in the guidance monitor that is continuously controlled to head for the player during billiard game. The predicted paths are calculated and projected by the beam projector onto the pool table bed in real time according to the change of the cue stick direction. The player could make a successful billiard shot conveniently comparing the predicted path with the solution paths. Through the ICBAS, more and more people are expected to enjoy billiards and be healthy. Furthermore, the proposed ICBAS can be applied to a humanoid robot player as well as a human player. A humanoid robot player has been already developed for playing simple Pool and Snooker. The ICBAS enables a humanoid player to play the more complicated Carom Billiards. In future work, we will expand the ICBAS to automatically determine the optimal solution path among several solution paths by considering the degrees of freedom and kinematic constraints of the humanoid robot player.

Declaration of conflicting interests
The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding
The author(s) disclosed the receipt of the following financial support for the research, authorship, and/or publication of this article: This study was supported by the Basic Science Research Programs (NRF-2019R1A6A1A09031717 and NRF-2018R1D1A1B07049270) through the National Research Foundation of Korea (NRF) funded by the Ministry of Education.

ORCID iD
Jaeyung Park https://orcid.org/0000-0002-5771-2872

Supplemental material
Supplemental material for this article is available online.

References
1. Hsu C-C, Tsai H-C, Chen H-T, et al. Computer-assisted billiard self-training using intelligent glasses. In: 2017 14th international symposium on pervasive systems, algorithms and networks and 2017 11th international conference on frontier of computer science and technology and 2017 Third international symposium of creative computing (ISPAN-FCST-ISCC), Exeter, UK, 21–23 June 2017.
2. Gao J, Zhu M, Liang H, et al. Design of the multiple neural network compensator for a billiard robot. In: Proceedings of 2015 IEEE 12th international conference on networking, sensing and control, Taipei, Taiwan, 9–11 April 2015.
3. Gao J, He Q, Zhan Z, et al. Dynamic modeling based on fuzzy neural network for a billiard robot. In: Proceedings of 2016 IEEE 13th international conference on networking, sensing and control, Mexico City, Mexico, 28–30 April 2016.
4. Fragiadaki K, Agrawal P, Levine S, et al. Learning visual predictive models of physics for playing billiards. In: Fourth international conference on learning representations (ICLR), San Juan, Puerto Rico, 2–4 May 2016.
5. Jebara T, Eyster C, Weaver J, et al. Stochastics: augmenting the billiards experience with probabilistic vision and wearable computers. In: International symposium on wearable computers, Cambridge, MA, USA, 13–14 October 1997, pp. 138–145.
6. Uchiyama H and Saito H. AR display of visual aids for supporting pool games by online markerless tracking. In: The 17th international conference on artificial reality and teleexistence, Esbjerg, Jylland, Denmark, 28–30 November 2007, pp. 172–179.
7. Shih C and Chu WC. A vision based interactive billiard ball entertainment system. In: The First IEEE international workshop on digital game and intelligent toy enhanced learning, Jhongli City, Taiwan, 26–28 March 2007, pp. 200–202.
8. Paolis LTD, Aloisio G, and Pulimeno M. A simulation of a billiards game based on marker detection. In: 2009 Second international conferences on advances in computer–human interactions, Cancun, Mexico, 1–7 February 2009, pp. 148–151.
9. Takahashi M, Kasai T, and Suzuki Y. Support system for pocket billiards. In: SICE annual conference 2008, Japan, 20–22 August 2008, pp. 3233–3236.
10. Ballard DHT. Generalizing the Hough transform to detect arbitrary shapes. Pattern Recogn 1981; 13(2): 111–122.
11. Yang J, Price B, Cohen S, et al. Object contour detection with a fully convolutional encoder–decoder network. In: 2016 IEEE conference on computer vision and pattern recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016.
12. Maninis K-K, Pont-Tuset J, Arbel´aez P, et al. Convolutional oriented boundaries: from image segmentation to high-level tasks. IEEE Trans Pattern Anal Mach Intell 2018; 40(4): 819–833.
13. Jian K and Gui S. Object and contour detection with an architecture-fusion network. In: 2021 IEEE 33rd international conference on tools with artificial intelligence (ICTAI), Washington, DC, USA, 1–3 November 2021.
14. Be Good at Billiards—the Diamond System, http://www3.sympatico.ca/eric.perreault/diamond_system_en.html#3 (1998, accessed 11 January 2019).
15. Balen J. A virtual billiard assistant. Master’s Thesis, University of Twente, 2009.
16. Li J, An X, and He H. Line segments detection with scale analysis: a principal component analysis based approach. In:
2009 IEEE international conference on intelligent computing and intelligent systems, Shanghai, China, 20–22 November 2009, pp. 43–47.

17. Lee S and Park J. Hand-eye LRF-based iterative plane detection method for autonomous robotic welding. Int J Adv Robot Syst 2015; 12(172): 1–14.

18. Camera Calibration Toolbox for Matlab, http://www.vision.caltech.edu/bouguetj/calib_doc (2015, accessed 11 January 2019).

19. Suzuki S and Abe K. Topological structural analysis of digitized binary images by border following. Comput Vis Graphics Image Process 1985; 30(1): 32–46.

20. Douglas D and Peucker T. Algorithms for the reduction of the number of points required for represent a digitized line or its caricature. Cartogr J 1973; 10: 112–122.

21. Sihombing DP, Nugroho HA, and Wibirama S. Perspective rectification in vehicle number plate recognition using 2D–2D transformation of planar homography. In: 2015 international conference on science in information technology, Yogyakarta, Indonesia, 27–28 October 2015, pp. 237–240.

22. Jain R, Kasturi R, and Schunck BG. Machine vision. New York, NY: McGraw-Hill Science, 1995.

23. Park JB, Lee SH, and Lee IJ. Precise 3D lug pose detection sensor for automatic robot welding using a structured-light vision system. Sensors 2009; 9: 7550–7565.

24. Chou C-W, Tien M-C, and Wu J-L. Billiards wizard: a tutoring system for broadcasting nine-ball billiards videos. In: 2009 IEEE international conference on acoustics, speech and signal processing, Taipei, Taiwan, 19–24 April 2009, pp. 1921–1924.

25. Landry J-F, Dussault J-P, and Mahey P. A heuristic-based planner and improved controller for a two-layered approach for the game of billiards. IEEE Trans Comput Intell AI Games 2013; 5(4): 325–336.

26. Fractional Aiming System, https://billiards.colostate.edu/faq/aiming/fractional (2014, accessed 12 August 2022).

27. Ho KHL, Martin T, and Baldwin J. Snooker robot player—20 years on. In: 2007 IEEE symposium on computational intelligence and games, Honolulu, HI, USA, 1–5 April 2007.

28. Pastor P, Kalakrishnan M, Chitta S, et al. Skill learning and task outcome prediction for manipulation. In: 2011 IEEE international conference on robotics and automation, Shanghai, China, 9–13 May 2011.

29. Nierhoff T, Leibrandt K, Lorenz T, et al. Robotic billiards: understanding humans in order to counter them. IEEE Trans Cybern 2016; 46(8): 1889–1899.