AR Display of Visual Aids for Supporting Pool Games by Online Markerless Tracking

Hideaki Uchiyama, Hideo Saito
Keio University, 3-14-1 Hiyoshi, Kohoku-ku, 223-8522, Japan
{*uchiyama, saito*}*@ozawa.ics.keio.ac.jp*

Abstract

This paper presents a supporting system for pool games by computer vision based augmented reality technology. Main purpose of this system is to present visual aids drawn on a pool table through LCD display of a camera mounted handheld device without any artificial marker. Since a pool table is rectangle, a pool ball is sphere and each has a specific color, these serve as a substitute for artificial markers. Using these natural features, the registration of visual aids such as shooting direction and ball behavior is achieved. Also, our supporting information is computed based on the rules of pool games and includes the next shooting way by simulating ball behavior. Experimental results represent that the accuracy of ball positions is enough for computing our supporting information.

Keywords: Computer Vision, Augmented Reality, Pool Games

1. Introduction

Pool game is one of popular sports in the world. A pool player uses a stick to hit the cue ball and pockets numbered balls into holes around the pool table. As is well known, playing pool games requires experienced skills and thought. It is necessary for pocketing balls to hit the cue ball precisely, and also to determine a direction and strength of a shot considering collisions and reflections. It is difficult for beginners to consider dynamic ball behavior, and this is one of key points for improving their skills.

There are two types of researches about pool games. One is pool playing robots [1, 2, 3, 4], and the other is supporting systems for beginners [5, 6]. In the robot researches, ball positions and the best shooting way are computed so that the robot plays pool. For estimating ball positions, these robots have the calibrated camera on the ceiling. For computing the shooting way, fuzzy logic [1], grey decision logic [3] and neural-fuzzy logic [4] are applied to determine the best shooting way based on the ball positions. In the supporting system researches, main purpose is to present visual aids for exercise. One uses a computer controlled laser pointer to project supporting information on a pool table [5], and the other uses a head mounted live video display (HMD) with a camera [6]. In the laser system, a camera and a laser system are set on the ceiling. Ball positions are estimated by the same way of those pool robots and visual aids are projected on the table by the laser. In the HMD system, ball positions are extracted from an image captured at an eye position assuming that a user sees the cue ball, a target ball and a target pocket. Since the camera is not calibrated, visual aids drawn on the HMD are just lines between ball and a pocket. In pool games, equipment such as a laser system and HMD are not desirable because installing the laser pointer system in a usual environment of a pool hall is hard, and playing pool with HMD is not natural for users. Also, visual aids are just for hitting the cue ball because these systems are designed for exercise.

This paper describes a pool supporting system that provides supporting information for games and presents visual aids drawn on the pool table through LCD display without artificial markers. This system is designed for Nine-Ball game which is one of popular pool games in the world and applied to a camera mounted handheld device such as a Tablet PC and a mobile phone. Before computing supporting information, ball positions are initially estimated [7]. As for computing supporting information, pool playing robots consider only initial ball positions, while our system considers not only initial ball positions but also ball positions after hitting the cue ball. This means that our supporting information includes the next shooting way. Finally, supporting information is drawn on captured images such as a kind of an augmented reality system. This concept is the same as that of the HMD system [6]. However, the registration of virtual objects is not achieved because the camera is not calibrated in the HMD system. In our system, this registration is achieved since the camera is calibrated by using natural features such as a pool table and pool balls.

2. Proposed System

2.1. System Overview

The usage and graphical user interface of our system are shown in Fig.1. First, a user standing beside the pool table captures the whole part of the table from an arbitrary viewpoint. Next, this system estimates ball positions from the captured image, and the user corrects ball positions by using the interface when this system does not estimate precisely ball positions. After getting ball positions, this system computes desirable shooting ways. When the user chooses one shooting ways, this system presents 2D and 3D supporting information such as ball behavior on the interface.

2.2. Algorithm Overview

In our system, it is important to estimate a projection matrix from an image which is captured from an arbitrary viewpoint for estimating ball positions and displaying 3D supporting information on the captured image. A projection matrix represents the relationship between an image coordinate system and a world coordinate system by the

DOI 10.1109/ICAT.2007.35

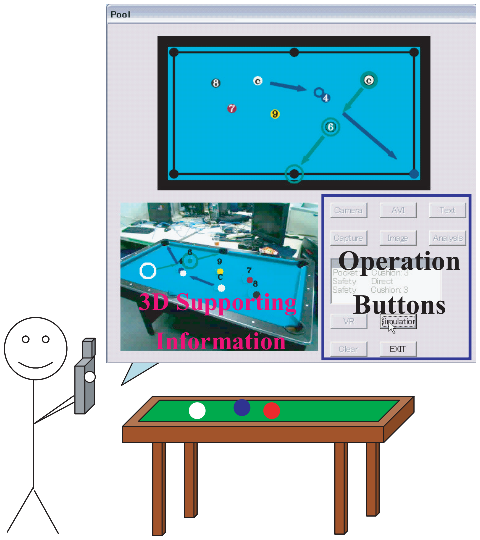


Figure 1: System Overview

following equation:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \sim \boldsymbol{P} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (1)$$

where (x, y) is an image coordinate and (X, Y, Z) is a world coordinate, $\boldsymbol{P}$ is a projection matrix. This system computes a projection matrix based on a homography that represents the case of $Z = 0$ in Eq.2 by using Simon's method [8, 9].

The algorithm is divided into three sections.

The first section is the estimation of ball positions on the pool table [7]. Since this estimation is based on the color segmentation, each color of the pool table and balls is measured as a preparation. First, a green planer area of the table is extracted from the captured image by color segmentation to compute four corner positions of the table in the image. The relationship of four corner positions between the captured image and the table provides a homography, which is transferred to a projection matrix by using Simon's method [8, 9]. Then, ball areas are also extracted from the image by color segmentation to compute a centre of each ball area, which can be transferred to the ball position on the table using the projection matrix. Finally, the ball number of each area is identified by a voting method.

The second section is the computation of shooting ways. Our system simulates ball behavior by giving several speeds to the cue ball to consider ball positions after hitting the cue ball. As for the result of each simulation, this system evaluates the result ball positions by using a cost function based on the distance and the angle of the balls and pockets. Some desirable shooting ways are selected in score order. It is important for beginners to plot strategies of winning the game by considering ball positions.

The last section is the presentation of supporting information. When a user chooses one shooting way, this system displays 2D supporting information and ball behavior on the interface. Also, this system provides 3D supporting information and ball behavior on the captured images while the user is capturing the pool table. This system calibrates the camera frame by frame online and generates background images by removing ball areas from the captured images. Then the 3D ball behavior is drawn on the background images while the user is capturing the table.

3. Estimation of Ball Positions

3.1. Estimation of Four Corner Positions

In order to obtain a homography for computing a projection matrix, this system estimates four corner positions of a pool table. An example of the images which is captured by a user standing besides the table is shown in Fig.2 (a). For estimating four corner positions of the green frame area, this system extracts the whole part of the green mat areas.

For RGB value of each pixel of the captured image, the inner angle with template color of the green mat is calculated. The angle is small in the case that two colors are similar, while the angle is large in the case that two colors are not similar. The mask of the green mat area shown in Fig.2 (b) can be generated by classifying by a threshold.

Next, four corners positions of the green frame area are estimated by computing four intersections of four line segments of the green frame area. Four line segments are computed from contours of the mask as shown in Fig.2 (c). This system applies Mata's method [10] which is implemented on OpenCV [11] for detecting line segments from contours. In the case that more than four line segments are detected as shown in Fig.2 (d), each pixel on all segments is projected to $\rho - \theta$ space that consists of a distance ρ from the centre of the table mask and an angle θ with x-axis. All samples in $\rho - \theta$ space are clustered based on the distance between samples. Each line segment of the green frame area is estimated by computing the centre of each cluster. Moreover, each segment is detected clockwise based on sorting by the values of θ because the origin in $\rho - \theta$ space is set on the centre of the captured image when the whole part of the table is included in the image. As shown in Fig.2 (e), the green mat area is extracted by computing the intersections of four line segments.

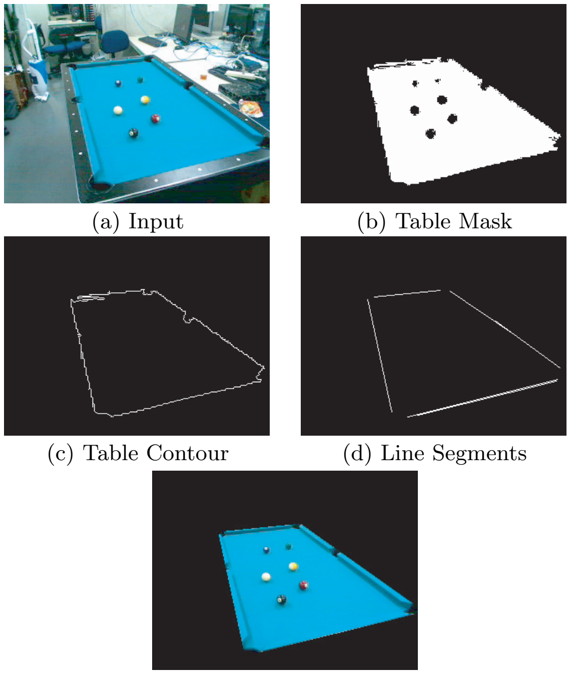
3.2. Camera Calibration

For calibrating a camera, this system computes a homography by applying Simon's method that provides a projection matrix from a homography. A homography is a matrix which represents 2D-2D relationship by the following equation:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \sim \boldsymbol{H} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \sim \boldsymbol{P} \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} \quad (2)$$

where $\boldsymbol{H}$ is a homography that is equal to the case of $Z = 0$ in a projection matrix. Since a homography is calculated using four correspondent points on the same plane, our system uses four corners of the green frame area. In Simon's method [8, 9], a projection matrix is calculated from a homography assuming that a skew component of the camera parameter is zero and a principal point is the image centre.

For calculating a homography, this system relates the corner positions in the image with the corresponding corner positions in a world coordinate system. Our system defines the left below corner in the captured image is related with


(a) Input (b) Table Mask

(c) Table Contour (d) Line Segments

(e) Table Area

Figure 2: Extraction of Four Corners

the origin in the world coordinate system as shown in Fig.3, and other corners in the image are related in a similar way.

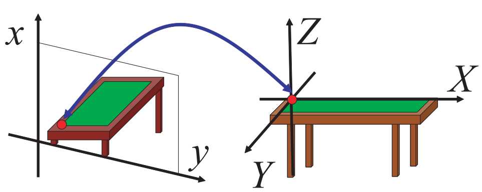


Figure 3: Relation of the Origin

3.3. Estimation of a Ball Position

In Eq.2, (X, Y) is a ball position on the pool table and Z-axis is vertical to the table. Since the plane of the homography is $Z = 0$ plane, Z value of the centre of the ball is determined as $Z = -(h - r)$ because the radius r of the ball and the height h of the cushion of the table are known. As shown in Fig.4, we consider the straight line that connects the centre of the ball (C) to the centre of the circle in the image (w). Then we define the point W at which the line passes through the ball surface. By getting (x, y) of point w in the image, (X, Y) of point C is computed.

3.4. Extraction of Ball Areas

To compute (x, y) of each ball in the captured image, this system extracts ball areas. Ball areas are extracted in the green mat area because balls exist on the area as shown in Fig.5 (a). For extracting the ball areas, non-green areas are extracted from the green mat area as candidates of the ball

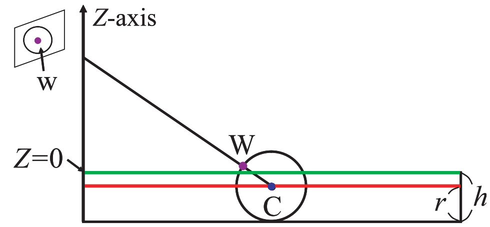
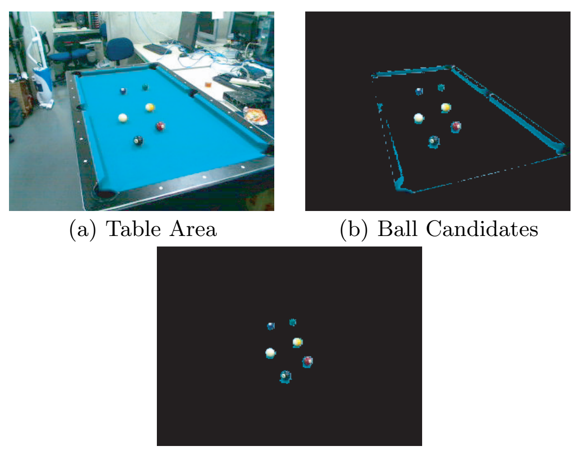


Figure 4: Estimation of a Ball Position

areas, and the ball areas are selected from the candidates. For RGB value of each pixel in the green mat area, the similarity score based on the inner product of the two color vectors and the difference of the two norms are computed. By classifying each pixel by a threshold, the candidates of the ball areas shown in Fig.5 (b) can be generated.

In the candidates of the ball areas, pocket areas and shadows of the cushion are included because these colors are also different from the color of the green mat. This system removes these areas depending on the size and the position of each area on the pool table. Since the size of the shadow area is larger than the size of a ball, the shadow area can easily be removed from the candidates by a threshold that is determined from the size of a ball area. The pocket area can also be removed from the candidates by calculating their positions from a projection matrix because the pocket area is out of the pool table area that the balls exist. Fig.5 (c) shows the result of Ball Areas.


(a) Table Area (b) Ball Candidates

(c) Ball Areas

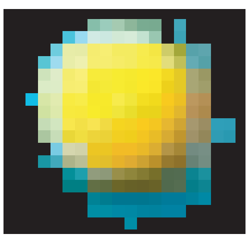
Figure 5: Extraction of Ball Areas

3.5. Identification of a Ball Number

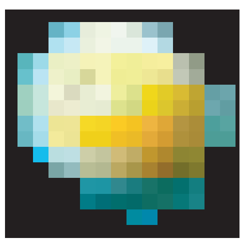
In pool supporting systems, it is important to identify a ball number as well as a position. Since our proposed system targets Nine-Ball game, this system should identify the ball numbers of white cue ball and from No.1 to No.9 ball as shown in Fig.6 (a). Two examples of the extracted ball areas are shown in Fig.6 (b) and Fig.6 (c). In addition to the ball color, green mat color and shadow are included in the area. This means that simple color threshold does not correctly identify the number of balls. Therefore, this system adopts the voting of the closest color pixel to the color template for each ball. As shown in Fig.6 (a), the color of each ball is measured beforehand as a template. In the rule of Nine-Ball game, the balls are one white cue ball, eight solid balls from No.1 to No.8 and one stripe No.9 ball. Since both No.1 ball and No.9 ball are yellow, white and the ball colors from No.1 to No.8 ball except the color of No.9 ball are used in this voting method.

For RGB value of each pixel in an extracted ball area, the inner angles with each template color are computed, and the counter of the color which gets minimum inner angle gets increased. This means that the counter of closest color with the color of each pixel gets increased. Table.1 shows two examples of the vote. As for No.9 ball which is one stripe ball, this ball is identified by the ratio of the sum of white and yellow. Other balls are identified by lion's share of votes. In Table.tb:vote, (b) is No.1 ball result and (c) is No.9 ball result.

(a) Template Colors



(b) No.1 Ball



(c) No.9 Ball

Figure 6: Identification of a Ball Number

Table 1: Classification by Voting Method

Number	c	1	2	3	4	5	6	7	8
(b)	25	106	0	0	0	28	27	5	21
(c)	56	42	0	0	0	17	16	0	15

3.6. Correction by User Interaction

In the case that some balls stand too close each other or close to the cushion, the ball areas sometimes cannot be extracted. For instance, the ball areas which are close to the cushion that is far side from the camera are not extracted in Fig.7 (the ball with yellow circle). Since the ball area is not extracted, the centre of the ball area which is necessary for computing a ball position is not computed. Therefore, the user inputs the centre of the ball area and the ball number by using the interface in this system. In Fig.7, a user clicks the centre of a ball area in the captured image, and the menu of ball numbers is popped up. After the ball number is selected by the user, this system estimates the ball position by using a projection matrix and displays the ball position on the right pool table image in the interface.

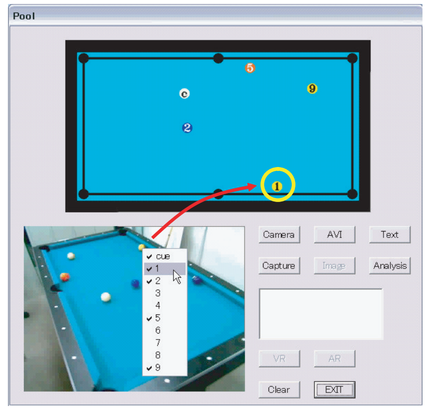


Figure 7: Correction by User Interaction

4. Computation of Shooting Ways

4.1. Rule Based Analysis

This system targets Nine-Ball game which is one of popular pool games in the world. The main rule of Nine-Ball game is that players have to pocket balls in an order of the ball's numbers. Also, the common rule of pool games is that a player should switch if a player cannot pocket any balls. Based on these two rules, it is necessary to consider shooting ways of Pocket and Safety to win the game.

Pocket means that a player aims at pocketing balls by considering a position of the next target ball if any ball can be pocketed, the player can hit the cue ball again. However, a player cannot always aim at pocketing a target ball because of some special ball arrangement. In the cases, a player aims at Safety which means that the player hits the cue ball not to pocket a target ball but to make ball arrangement hard to aim at a target ball for other player. This is from the second rule as mentioned above that a player should switch with the next player if the player can not pocket any ball. In Nine-Ball game, it is important for strategies for winning the game to judge whether a player can pocket a target ball or not and to consider the shooting way of aiming at Safety. For this reason, our proposed system searches some shooting ways of Pocket and Safety by simulating ball behavior. In addition, Free Ball is also one of important events in pool games, which occurs in the case that an adversary player fouls. In the case of Free Ball, a player can put the cue ball anywhere. Thus, our system simulates ball behavior assuming that the cue ball is put on several places. As for simulation of ball behavior, this system assumes that ball behavior follows particle dynamics and linear motion. Also, this system assumes that a player hits the centre of the cue ball.

The algorithm overview is as follows. First, this system searches ball paths of Pocket and Safety for aiming at a target ball directly or using a cushion. Each found path is evaluated based on an angle and distances as shown in Fig.8 and a basic cost function is as follow:

$$E = a\,(1 - \cos\theta) + b \prod_i \frac{1}{d_i} \tag{3}$$

where a and b are weighting factors for distances and angles. The score E is higher in the case that each distance is shorter and the angle is closer to 180 degree. The score is zero if another obstacle ball is on a ball path as shown in Fig.8. Next, this system simulates ball behavior by giving several speeds to the cue ball. Ball positions of each simulation result are evaluated by the same way as shown in Fig.8 and Eq.3. This system calculates some desirable shooting ways based on the two scores.

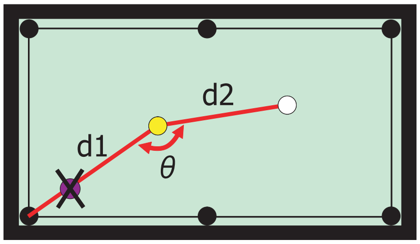
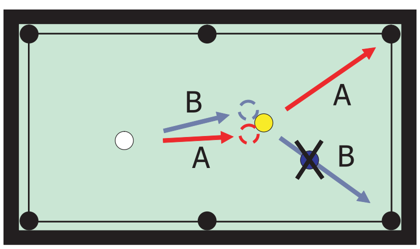


Figure 8: Evaluation of Ball Arrangement

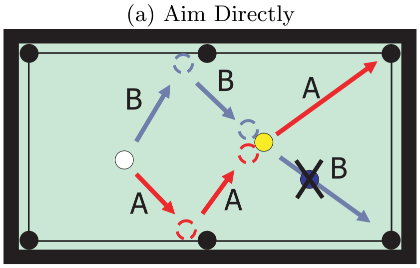
4.2. Pocket

Pocket means that a player aims at pocketing a target ball by considering a position of a next target ball. Fig.9(a) represents an example of aiming at a target ball directly and Fig.9(b) represents an example of aiming at a target ball using a cushion.

In Fig.9(a) and Fig.9(b), there are two ball paths in each shooting way. This system does not simulate ball behavior on ball path B because of an obstacle ball. This system also searches the best speed in each ball path by giving several speeds to the cue ball in computer simulations.



(a) Aim Directly



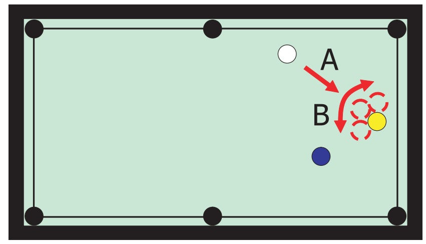
(b) Aim using a Cushion

Figure 9: Pocket

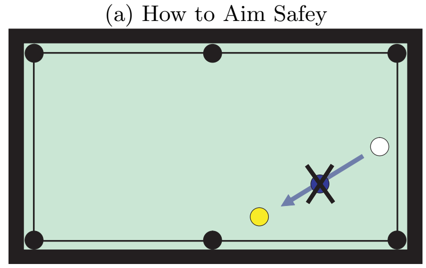
4.3. Safety

Safety also means that a player hits the cue ball not to pocket a target ball but to make ball arrangement hard to aim at a target ball for the next player.

In Fig.10(a), it is difficult to pocket a target ball. For this ball arrangement, a player should aim at Safety. In the case that ball arrangement is like Fig.10(b), next player cannot aim at a target ball directly. Thus, ball arrangement which score is zero as mentioned above is a desirable Safety and Fig.10(b) represents successful Safety. For searching, this system simulates ball behavior assuming that a player aims at the neighborhood of a target ball as shown in Fig.10(a).



(a) How to Aim Safey



(b) Result of (a)

Figure 10: Safety

4.4. Free Ball

Free Ball is that a player can put the cue ball anywhere when an adversary player fouls. For simulation of Free Ball, this system simulates ball behavior by the same way as Pocket assuming that the cue ball is put on the circumference of a target ball as shown in Fig.11.

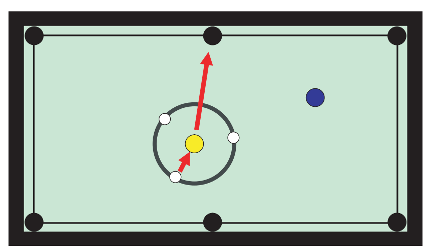
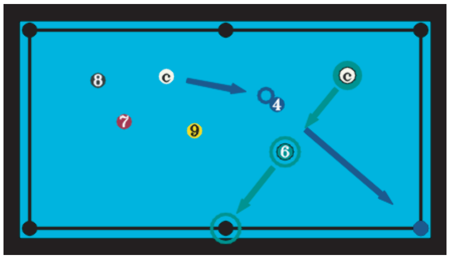


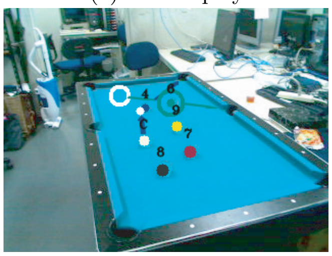
Figure 11: Free Ball

5. Presentation of Supporting Information

Our proposed system has two types of a method to display supporting information. Fig.12 (a) represents 2D supporting information, which is called VR display. Fig.12 (b) is a kind of an augmented reality system that 3D supporting information such as balls and arrows are drawn on the images that a user is capturing, which is called AR display.



(a) VR Display



(b) AR Display

Figure 12: VR/AR Display

For the AR display, a background image is generated from a captured image by replacing ball areas with the color of a green mat color as shown in Fig.13.

(a) Input

(b) Removal of Balls

Figure 13: Background

Our proposed system assumes that a ball is approximated to a circle in the image for drawing a ball. A radius R of a circle is calculated by the ratio of the focal length f from a projection matrix and the distance D between a camera and a ball as shown in Fig.14 and by the following equation:

$$r = \frac{fR}{D} \tag{4}$$

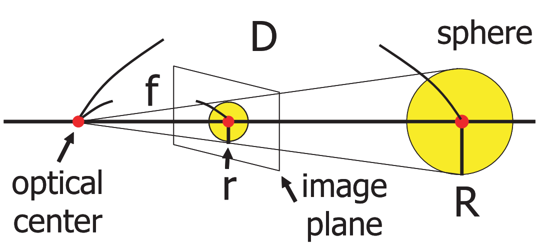
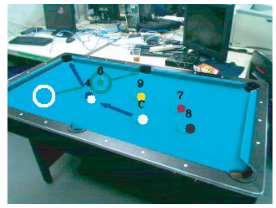
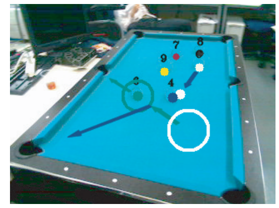


Figure 14: Estimation of a ball radius in an image

3D supporting information in this AR display is based on online camera calibration by the same way as Sec.3.2 and drawn on the image by CG. As for determining the origin of the world coordinate system, the method of Sec.3.2 is different from the method of AR display. In Sec.3.2, left below corner is defined as the origin, while the origin is determined by using computed ball positions because each corner has possibilities to be the origin. Fig.15 represents that a user see 3D supporting information moving around the table.



(a) Viewpoint1



(b) Viewpoint2

Figure 15: Online AR Display

6. Experimental Results

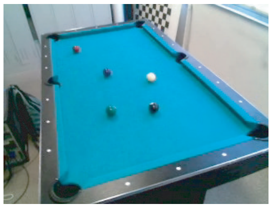
6.1. Accuracy of Ball Positions

For evaluating the accuracy of ball positions, the error between the positions which are measured beforehand and the results of our proposed system is calculated. The size of the pool table is $1330mm \times 700mm$, the radius of a ball is $24mm$ and the size of a captured image is $320 \times 240 pixel$. Also, the CPU is Pentium M(1.6GHz), and the RAM capacity is 512MB.

Fig.16(a) is a captured image. This system estimates ball positions from this image for less than one second. The extracted green mat area is shown in Fig.16(b). Also, the ball areas is shown in Fig.16(c). Fig.16(d) represents the comparison between the positions which are measured beforehand and the results of this system. Square dots and circular dots denote the positions which are measured beforehand and the result detected positions by this system respectively.

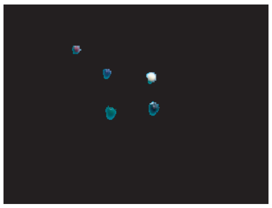
Table.2 represents the comparison between positions measured beforehand and positions estimated by our proposed system. The maximum error is $9mm$, and the average of errors is $6mm$. We consider that this error is mostly caused by the following two reasons. First, our proposed system assumes that a principal point is approximated to the center of a captured image. Second, the center coordinate of a ball area is not precise because the ball area includes not only the ball color area but also the border with cushion. However, the average error is less than 1% of the short side length of the pool table which is $700mm$, so that we can say that the accuracy of this system is sufficient to provide appropriate supporting information to the users.

For evaluating the stability of the estimated results by changing viewpoints, ball positions are estimated from 120 images. Fig.17 represents the comparison between each estimated results and the averages of all estimated results. The average of the difference between each results and the averages of all estimated results is $4mm$, and the error is attributed to the lighting condition depending on the view direction.

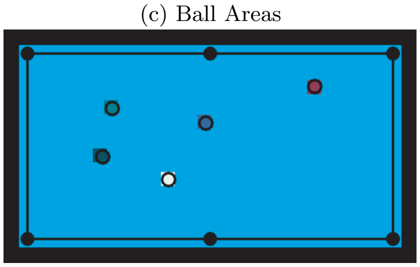


(a) Input

(b) Table Area



(c) Ball Areas



(d) Comparison Between Ground Truths and Results

Figure 16: Accuracy of Ball Positions

Table 2: Estimated Results(mm)

Number	8		6	
Coordinate	x	y	x	y
Result	388	280	214	314
Ground truth	388	289	219	323
Number	c		4	
Coordinate	x	y	x	y
Result	470	511	270	641
Ground truth	465	519	271	650
Number	7			
Coordinate	x	y		
Result	149	1031		
Ground truth	143	1028		

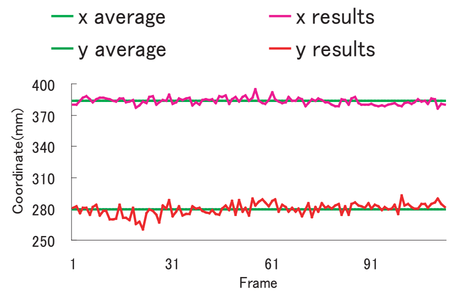


Figure 17: Estimated Results of No.8 Ball in Each Image

6.2. Estimation of a Camera Position

A camera position is estimated from a rotation matrix $\boldsymbol{R}$ and a translation matrix $\boldsymbol{t}$ computed from a projection matrix by the following equation:

$$\begin{pmatrix} X_c \\ Y_c \\ Z_c \end{pmatrix} = \boldsymbol{R} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} + \boldsymbol{t} \tag{5}$$

where (X, Y, Z) is an world coordinate, and (X_c, Y_c, Z_c) is a camera coordinate. $(0, 0, 0)$ in a camera coordinate system correspond to a camera position in an world coordinate system. Also, the vector of $(0, 0, 1)$ in a camera coordinate system represents a camera direction. Fig.18 shows a camera position of Fig.16 (a), and the red frames of a four-sided pyramid is a camera in Fig.18.

7. Discussions

For evaluating the usability of the system, this system is used by seven beginners and gets some comments.

As a experiment of Free Ball which is one of important events in pool games, the beginners saw an example of ball arrangement as shown in Fig.19(a) and considered where to put the cue ball Our proposed system provides several supporting information as shown in Fig.19(b),(c),(d). Supporting information is that arrow A is a ball path of first shot, point F is a desirable arrival point where the cue ball stopped after a player hits the cue ball, and arrow B is a ball path of next shot when a player can put the cue ball near the point F.

Fig.19(b) represents the caution of pocketing the cue ball which is one of fouls in pool games. If a player puts the cue ball on the place as shown in Fig.19(b) and hits the cue ball linearly, the cue ball may be put on the pocket C. This information is one of advantages by simulating ball behavior. Fig.19(c),(d) represent the position of the cue ball and the shooting ways of pocketing a target ball considering the position of a next target ball.

Beginners said that it is helpful to display several shooting ways. For beginners, they sometimes find only one shooting way and the shooting way is not always best way. Also, they hardly consider the shooting way of Safety. In this system, they can select one desirable way from several shooting ways for them, and they can learn so much about various shooting ways.

In addition, it is also helpful to see ball behavior before shooting the cue ball. Since they can intuitively understand 2D and 3D ball behavior, they can rehearse the shooting way in their head.

As an improvement, our proposed system cannot teach the shooting ways of jump shot and curled shot since this system assumes that ball behavior follows particle dynamics and linear motion. In the future, this system will also simulate ball behavior based on rotational dynamics.

8. Conclusions

We have proposed a system for supporting pool game based on analysis of ball positions for Nine-Ball Game. A user can capture a pool table from an arbitrary viewpoint, and see supporting 3D information and ball behavior which are drawn on the captured images through LCD display. In this system, ball positions are estimated by calculating a pro-

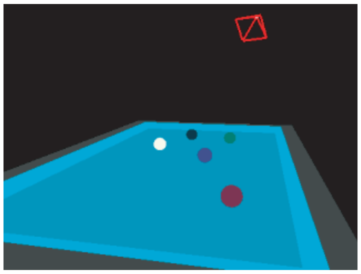


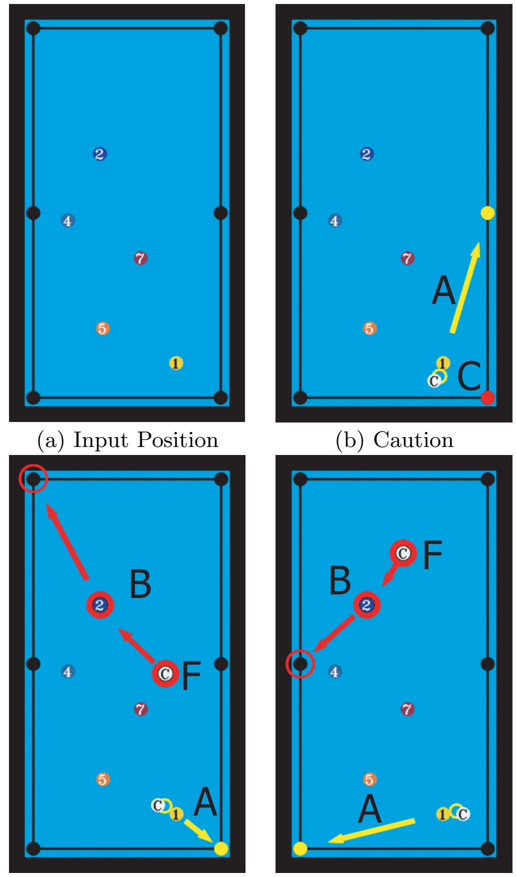
Figure 18: Estimation of a Camera Position

jection matrix from the captured image from an arbitrary viewpoint. A voting based method is applied to identify a ball number of a ball area to avoid effects of a cushion color, shadows and specular reflection. As for supporting information, this system simulates ball behavior by giving several speeds to the cue ball and calculates some desirable shooting ways based on the rule of the pool game. Moreover, supporting information and 3D ball behavior are online drawn on the images while the user is capturing the pool table. In the experimental results, the accuracy of estimated ball positions is enough for analysis of ball arrangement.

In the future, we would like to apply this system to a camera mounted mobile phone as one of portable device.

Acknowledgment

This work is supported in part by a Grant-in-Aid for the Global Center of Excellence for High-Level Global Cooperation for Leading-Edge Platform on Access Spaces from the Ministry of Education, Culture, Sport, Science, and Technology in Japan.



(a) Input Position (b) Caution (c) Shooting Way 1 (d) Shooting Way 2

Figure 19: Supporting Information

References

[1] E.K.Wong S.C.Chua and V.C.Koo. Pool balls identification and calibration for a pool robot. In *Proc. of the International Conference on Robotics, Vision, Information and Signal Processing*, pages 312–315, 2003.

[2] M.T.M.Shalmani P.Karimian M.E.Alian, S.B.Shouraki and P.Sabzmeydani. Roboshark: a gantry pool player robot. In *Proc. of the International Symposium on Robotics*, 2003.

[3] J.S.Yang Z.M.Lin and C.Y.Yang. Grey decision-making for a billiard robot. In *Proc. of the International Confrence on Systems, Man and Cybemetrics*, volume 6, pages 5350–5355, 2004.

[4] J.Li B.Cheng and J.Yang. Design of the neural-fuzzy compensator for a billiard robot. In *Proc. of the International Confrence on Networking, Sensing and Control*, volume 2, pages 909–913, 2004.

[5] K.L.Jensen L.B.Larsen, R.B.Jensen and S.Larsen. Development of an automatic pool trainer. In *Proc. of the ACM SIGCHI International Conference on Advances in computer entertainment technology*, pages 83–87, 2005.

[6] J.Weaver T.Starner T.Jebara, C.Eyster and A.Pentland. Stochastics: Augmenting the billiards experience with probabilistic vision and wearable computers. In *Proc. of the International Symposium on Wearable Computers*, pages 138–145, 1997.

[7] H.Uchiyama and H.Saito. Position estimation of solid balls from handy camera for pool supporting system. In *Proc. of the Pacific-Rim Symposium on Image and Video Technology*, volume 4319 of *Lecture Notes in Computer Science*, pages 393–402. Springer, 2006.

[8] G.Simon and M.Berger. Reconstructing while registering: a novel approach for markerless augmented reality. In *Proc. of the International Symposium on Mixed and Augmented Reality*, pages 285–294, 2002.

[9] Y.Uematsu and H.Saito. Vision-based registration for augmented reality with integration of arbitrary multiple planes. In *Proc. of the International Conference on Image Analysis and Processing*, pages 155–162, 2005.

[10] C.Galambos J.Matas and J.Kittler. Progressive probabilistic hough transform for line detection. In *Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 554–560, 1999.

[11] Intel. `http://www.intel.com/`.