

## Research Article

# AR Supporting System for Pool Games Using a Camera-Mounted Handheld Display

**Hideaki Uchiyama and Hideo Saito**

*Faculty of Science and Technology, Keio University, Hiyoshi Kohoku-ku, Yokohama-shi, Kanagawa-ken 223-8522, Japan*

Correspondence should be addressed to Hideaki Uchiyama, uchiyama@ozawa.ics.keio.ac.jp

Received 2 October 2007; Revised 21 July 2008; Accepted 28 August 2008

Recommended by Adrian Cheok

This paper presents a pool supporting system with a camera-mounted handheld display based on augmented reality technology. By using our system, users can get supporting information when they once capture a pool table. They can also watch visual aids through the display while they are capturing the table. First, our system estimates ball positions on the table with one image taken from an arbitrary viewpoint. Next, our system provides several shooting ways considering the next shooting way. Finally, our system presents visual aids such as shooting direction and ball behavior. Main purpose of our system is to estimate and analyze the distribution of balls and to present visual aids. Our system is implemented without special equipment such as a magnetic sensor or artificial markers. For evaluating our system, the accuracy of ball positions and the effectiveness of our supporting information are presented.

Copyright © 2008 H. Uchiyama and H. Saito. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

## 1. Introduction

A pool game is one of popular sports in the world and enjoyed by people of all ages. Pool players hit the cue ball by using a long pole and pocket numbered balls into holes around a pool table. As being well known, pocketing balls requires much skills and consideration. It is necessary not only to hit the cue ball precisely, but also to consider the direction and strength of the shot. Especially, it is difficult for beginners to hit the cue ball considering laws of physics such as collisions and reflections.

There are two types of studies about pool games. First type includes studies such as developing a robot that plays pool [1–5], and second type includes studies such as developing a supporting system for beginners [6, 7].

In the studies for developing pool-playing robots, main research topic is that a robot placed above a pool table automatically recognizes ball positions and decides the best shooting way by analyzing positional relationship between pockets and balls for playing pool with human beings. As for recognition of balls, a camera should be hung exactly above the center of the table which provides the sensory input for these robots. Each system estimates ball positions

by using color identification with the calibrated camera. The difference among these researches is computation of best shooting way based on analyzing the distribution of balls. Alian et al. [1] used a cost function composed of distances and angles to compute the strength and the direction of the shot and a fuzzy approach to select the best shot. Lin et al. [2] adopted grey decision making which is similar with fuzzy algorithm and can deal with uncertainty and incompleteness of the pool game to find the best shot. Cheng et al. [3] also used the same fuzzy approach as Alian [1], however, Cheng's algorithm included the predictable hitting error model which was developed based on the recorded database of pocketing processes. Chua et al. [4, 5] used another fuzzy approach which was called Sugeno Fuzzy and evaluated the effectiveness of the selected shot based on fuzzy logic.

In the studies on supporting system, main research topic is the way of presenting of visual aids for practice. Jebara et al. [6] proposed an augmented reality-based wearable computer system which used a head-mounted live video display with a camera. However, their visual aids were just drawn lines and the registration of visual aids was not achieved because the camera was not calibrated. They suggested that it was necessary to calibrate the camera to display more effective

visual aids. Larsen et al. [7] proposed a framework of an automatic pool trainer. A camera was set above the pool table to estimate ball positions by the same way as the studies for the pool-playing robots and a projector was also set for visual output on the table to users. In addition, users can issue spoken commands to an interface agent acting as a personification of the system.

For a pool-supporting system for games, it is necessary to consider not only the computation of shooting ways suggested in the studies for the pool-playing robots but also the display of visual aids suggested in the studies for pool-supporting systems. As for computation of shooting ways, only current distribution of balls is analyzed in the studies for the robots. However, it is important for pool players to analyze the next distribution of balls after the shot because they can hit the cue ball twice in a row after pocketing a target ball. The players should also consider next strategy such as Pocket that means putting a ball into a pocket or Safety that means disturbing the adversary player to put a ball into a pocket. As for display of visual aids, a head-mounted live video display is not appropriate because it is not natural for players. It is also difficult to install the projector system in a usual environment of a pool hall because the system is huge and takes much cost. It is ideal that special equipment is not utilized.

In this paper, we propose a pool-supporting system that solves these problems as mentioned above. We adopt a camera-mounted handheld display such as a mobile phone because, such device is small and widespread. Our system provides supporting information for games considering the strategies and the next distribution of balls. It is important for beginners to plot strategies of winning the game by the next distribution of balls. The supporting information is presented as visual aids through the display. In our system, ball positions are initially estimated [8]. Then, supporting information for several strategies are computed by simulating ball behavior. Finally, visual aids are drawn on captured images based on a kind of an augmented reality system [9] (this concept is the same as that of the Jebara's system [6]); but they did not draw visual aids and did not register them because the camera was not calibrated in their system. In our system, this registration is achieved since the camera is calibrated by using natural features such as a pool table and balls.

## 2. Proposed System

### 2.1. System Overview

The usage and graphical user interface of our prototype system are shown in Figure 1, which is implemented on a Tablet PC that is similar with a mobile phone. Our system is designed for Nine-Ball game which is one of popular pool games in the world among others.

First, a user standing beside the pool table captures the whole part of the table from an arbitrary viewpoint. Next, our system estimates ball positions from the captured image. The user corrects ball positions by using the interface if our system does not precisely estimate ball positions. After

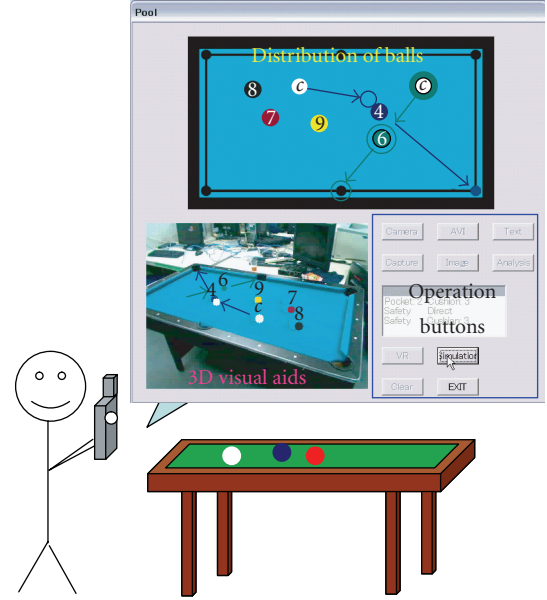


FIGURE 1: System overview.

getting ball positions, our system computes some shooting ways for each strategy. Since our system provides several shooting ways, users can select one of the ways based on ability, and learn the shooting ways that they cannot notice. When the user chooses one shooting way, the user can watch visual aids such as direction of the shot and ball behavior from an arbitrary viewpoint. To sum up the matter, the user can get supporting information when the user once captures the table. The user can also watch visual aids through the display while the user is capturing the table.

### 2.2. Algorithm Overview

In our system, it is important to estimate a projection matrix of an image which is captured from an arbitrary viewpoint for estimating ball positions and displaying visual aids on the captured image. A projection matrix represents the relationship between an image coordinate system and a world coordinate system by the following equation:

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

where  $(x, y)$  is an image coordinate and  $(X, Y, Z)$  is a world coordinate,  $\mathbf{P}$  is a projection matrix. Our system computes a projection matrix based on a homography that represents the case of  $Z = 0$  in (1) by using Simon's method [10, 11].

The algorithm is divided into three sections.

The first section is estimation of ball positions on the pool table [8]. Ball position estimation is based on camera calibration using planar area. First, a green planar area of the table is extracted from a captured image by color segmentation to compute four corner positions of the table in the image. The relationship of four corner positions

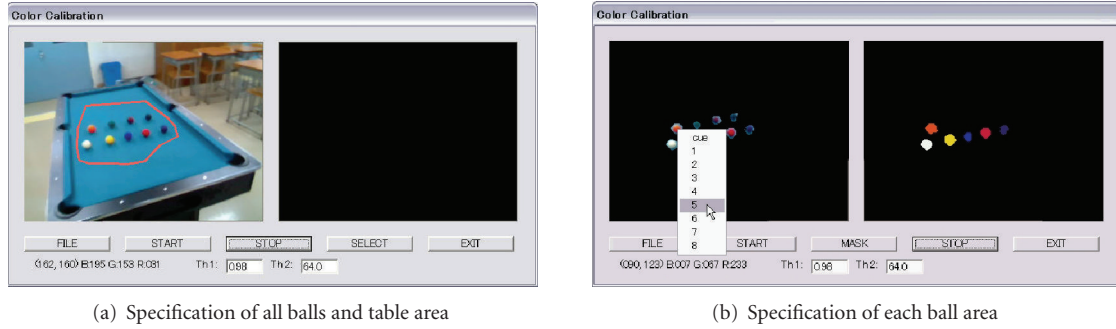


FIGURE 2: Color acquisition.

between the image and the table provides a homography, which is transferred to a projection matrix by using Simon's method [10, 11]. Then, ball areas are also extracted from the image by color segmentation to compute the center of each ball area, which can be transferred to the ball position on the table using the projection matrix. Finally, a ball number of each area is identified by a voting method.

The second section is computation of shooting ways. Our system considers some strategies based on the rule of a pool game and the next shooting way after a shot. First, our system finds ball paths of each strategy in which a player can physically hit the cue ball at a target ball. In each ball path, our system simulates ball behavior by giving several speeds to the cue ball to consider next distribution of balls after hitting the cue ball. Also, each ball path is evaluated by a cost function based on a distance and an angle. As a result, some desirable shooting ways are selected in score order.

The last section is presentation of visual aids [9]. When a user chooses one shooting way, our system displays visual aids such as a direction of the shot and ball behavior on the captured images while the user is capturing the pool table. Our system calibrates the camera frame by frame online and generates background images by removing ball areas from the captured images. Then, the ball behavior is drawn on the background images while the user is capturing the table. Since our system can draw any 3D objects such as a performance of trained player on the image by using this algorithm, our system can draw more plentiful and precise visual aids than those of a previous research [6].

### 2.3. Preparation for the System

The initial inputs of our system are the sizes of a pool table and balls and the colors of balls. The sizes of a pool table and balls are determined in advance for official rules and are not necessary to be measured. On the other hand, the colors of balls depend on lighting conditions of the room. For this reason, the colors are needed to be measured once.

Figure 2 represents the graphical user interface of the color acquisition system. The mat area seems to be blue in all figures because of color response of our camera. The mat area of the pool table actually seems to be green. For this reason, we use "green" uniformly to indicate the color of the pool table.

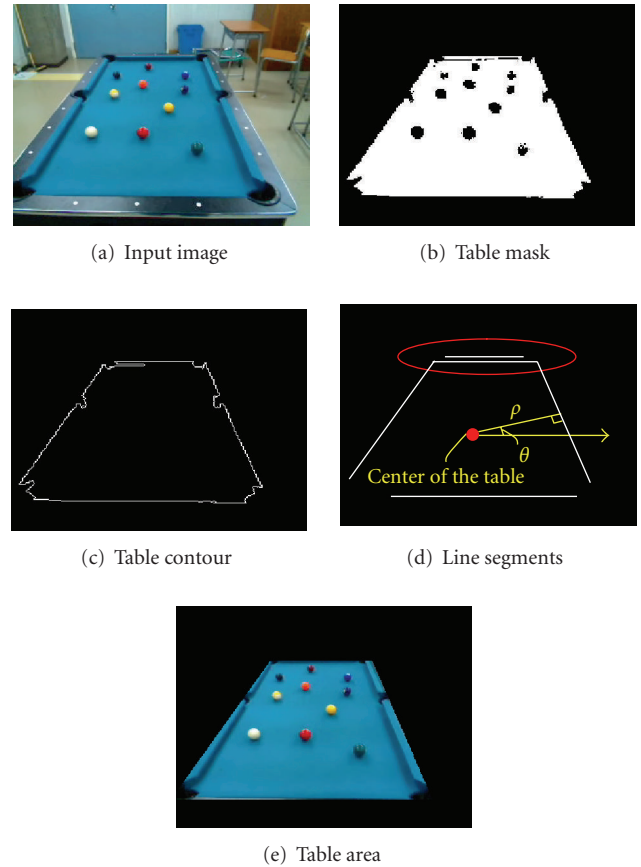


FIGURE 3: Extraction of four corners.

The input image of our system needs to include nine solid balls and a pool table. A user segments an area that includes the balls and the table as shown in Figure 2(a). Our system initially decides the color of the table by making RGB histogram of the segmented area and computing a peak of the histogram. Then, our system extracts ball areas by removing the table area, assuming that ball colors are not the same as the color of table. Finally, a user clicks each ball area as shown in Figure 2(b), and our system outputs the colors of the balls and the table as a text file. Since this acquisition of the colors is once measured beforehand if the lighting condition is constant, a user does not necessarily have to measure the

colors and another person measures these parameters in a usual environment of pool games.

### 3. Estimation of Ball Positions

#### 3.1. Estimation of Four Corner Positions

In order to obtain a homography for computing a projection matrix, our 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 Figure 3(a). For estimating four corner positions of the green frame area, our system extracts whole part of the green mat areas.

For RGB value of each pixel in the captured image, the inner angle of the two color vectors is calculated by the following equation:

$$\text{Angle} = \arccos \left( \frac{\mathbf{C}_t \cdot \mathbf{C}_p}{|\mathbf{C}_t| |\mathbf{C}_p|} \right), \quad (2)$$

where  $\mathbf{C}_t$  is the template color as mentioned above,  $\mathbf{C}_p$  is each pixel color, and Angle is the angle between the vectors  $\mathbf{C}_t$  and  $\mathbf{C}_p$ . The angle is small if two colors are similar, while the angle is large if two colors are not similar. Each pixel is classified by setting a threshold on 10 degree that is determined via experiments. As a result, a mask of the green mat area shown in Figure 3(b) can be generated.

Next, four corners positions of the green frame area are estimated by calculating four intersections of four line segments of the green frame area. Four line segments are computed from contours of the mask as shown in Figure 3(c). Our system applies Mata's method [12] which is implemented on OpenCV [13] for extracting line segments from contours.

In Figure 3(d), more than four line segments are extracted and these segments are clustered into four segments. The space for the clustering is  $\rho - \theta$  space denoted in Figure 3(d). For each line, the perpendicular line from the center of the green table area is connected and the connected point is calculated.  $\rho$  is the distance between the center and the connected point and  $\theta$  is the angle between  $x$ -axis of the image and the vector connecting the center and the connected point. In the space, line segments have the values of  $\rho$  and  $\theta$ , and they are clustered into four clusters by using  $k$ -means clustering method ( $k = 4$ ). In Figure 3(d), two lines in the ellipse become one cluster and the center of the cluster is calculated for one line. In addition, these lines can be extracted clockwise based on the value of  $\theta$ .

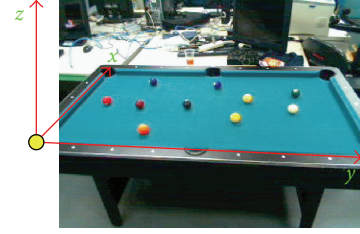
As shown in Figure 3(e), the green mat area is extracted by calculating the intersections of four line segments.

#### 3.2. Camera Calibration

For calibrating a camera, our system computes a homography by applying Simon's method [10, 11] that provides a projection matrix from a homography.



(a) Short side



(b) Long side

FIGURE 4: Definition of a world coordinate system.

A homography is a matrix which represents 2D-2D relationship by the following equation:

$$\mathbf{H} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}. \quad (3)$$

A homography is computed from four corresponding points in the same plane because of eight degrees of freedoms.

The relationship between a homography and a projection matrix is represented by the following equation:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \sim \mathbf{H} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \sim \mathbf{P} \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix}. \quad (4)$$

A homography is equal to the case that a  $Z$  component is 0 in a projection matrix which represents 2D-3D relationship.

However, (5) represents components of a projection matrix, and (6) represents components of a homography.  $\mathbf{A}$  consists of intrinsic parameters,  $\mathbf{r}_i$  is a component vector of a rotation matrix, and  $\mathbf{t}$  is a translation matrix.  $[\mathbf{r}_1 \mathbf{r}_2 \mathbf{r}_3 \mathbf{t}]$  represents the rigid transform that consists of extrinsic parameters. As mentioned above, a homography is equal to the case that  $\mathbf{r}_3$  is not included in a projection matrix:

$$\mathbf{P} = \mathbf{A}[\mathbf{R}|\mathbf{t}] = \mathbf{A}[\mathbf{r}_1 \mathbf{r}_2 \mathbf{r}_3 \mathbf{t}], \quad (5)$$

$$\mathbf{H} = \mathbf{A}[\mathbf{r}_1 \mathbf{r}_2 \mathbf{t}]. \quad (6)$$

For computing a projection matrix from a homography,  $\mathbf{r}_3$  is calculated by the cross product of  $\mathbf{r}_1$  and  $\mathbf{r}_2$  which is one of properties of a rotation matrix by the following equation:

$$\mathbf{r}_3 = \mathbf{r}_1 \times \mathbf{r}_2. \quad (7)$$

For calibrating the camera, our system computes a homography by relating four corner positions in the image



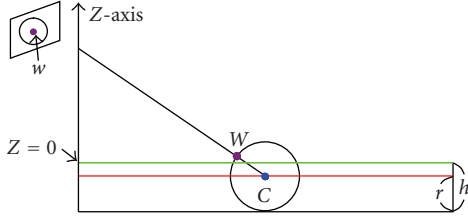


FIGURE 5: Estimation of a ball position.

with the corresponding corner positions in a world coordinate system. For computing a homography, it is necessary to define a world coordinate system. As shown in Figure 4, there are two probabilities of a side on which a user captures the table. In the case of the short side, the difference between the length of the short side and the long side is small. On the other hand, the difference is large in the case of the long side. By judging the side based on a threshold (in our case, 100 pixels), our system initially identifies on which side a user captures the table. Then, our system defines a world coordinate system as shown in Figure 4(a) or 4(b). To compute a homography, the left below corner in the captured image is related with the origin in the world coordinate system, and other corners in the image are related in a similar way.

Next,  $\mathbf{A}$  and  $[\mathbf{r}_1 \mathbf{r}_2 \mathbf{t}]$  are computed from a homography. A homography is represented by the following equation:

$$\mathbf{H} = \begin{bmatrix} f & 0 & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & t_1 \\ r_{21} & r_{22} & t_2 \\ r_{31} & r_{32} & t_3 \end{bmatrix}, \quad (8)$$

where  $f$  is a focal length and  $(c_x, c_y)$  is a principal point. Our system assumes that  $(c_x, c_y)$  is approximated to a center of a captured image and obtained depending on the size of a captured image. Also, our system applies one of properties of a rotation matrix by the following equation:

$$\mathbf{r}_1 \cdot \mathbf{r}_2 = 0. \quad (9)$$

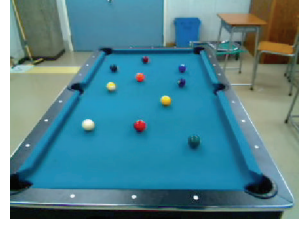
Through calculation processing, our system computes a projection matrix from a captured image.

### 3.3. Estimation of a Ball Position

In (4),  $(X, Y)$  is a ball position on the table and  $Z$ -axis is vertical to the table. Since the plane of the homography is  $Z = 0$  plane,  $Z$  of the center 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 Figure 5, we consider the straight line that connects the center of the ball ( $C$ ) to the center 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

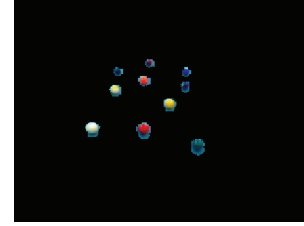
To compute  $(x, y)$  of each ball in the captured image, our system extracts ball areas. Ball areas are extracted in the green



(a) Table area



(b) Ball candidates



(c) Ball areas

FIGURE 6: Extraction of ball areas.

mat area because balls exist on the area. For extracting the ball areas, non-green areas are extracted from the green mat area as candidates of the ball 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 following equation:

$$E = a \frac{\mathbf{C}_t \cdot \mathbf{C}_p}{|\mathbf{C}_t| |\mathbf{C}_p|} + b \left( 1 - \frac{|\mathbf{C}_t - \mathbf{C}_p|}{C_{\text{MAX}}} \right), \quad (10)$$

where  $\mathbf{C}_t$  is the template color,  $\mathbf{C}_p$  is the color of each pixel, and  $E$  is the score which is composed of the inner product and the difference of the norms. By setting  $a$  on 0.7 and  $b$  on 0.3, each pixel is classified by setting a threshold on 0.8, and the candidates of the ball areas shown in Figure 6(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. Our 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 using a threshold that is determined from the size of a ball area (from 50 to 400 pixels). 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 area where balls exist.

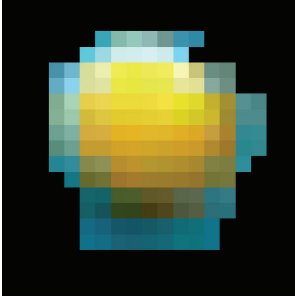
Figure 6(c) shows the result of ball areas.

### 3.5. Identification of a Ball Number

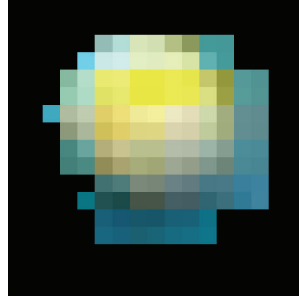
In pool-supporting systems, it is important to identify a ball number as well as a position. Since our proposed system targets NineBall game, our system should identify the ball



(a) Template colors



(b) Ball number 1



(c) Ball number 9

FIGURE 7: Identification of a ball number.

numbers of white cue ball and balls from number 1 to number 9 as shown in Figure 7(a).

Two examples of the extracted ball areas are shown in Figures 7(b) and 7(c). In addition to the ball color, green mat color and shadow are included in the area. Our system identifies the ball number based on the voting of the closest color pixel to the color template for each ball.

As shown in Figure 7(a), the color of each ball is measured beforehand. In the rule of NineBall game, balls are one white cue ball, eight solid ball from number 1 to number 8 and one stripe number 9 ball. Since both ball number 1 and ball number 9 are yellow, white and the colors from ball number 1 to ball number 8 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. Two examples of the vote are shown in Tables 1(b) and 1(c).

First, ball number 9 which is one stripe ball is identified by the following equation:

$$\frac{b_c + b_1}{b_{\text{sum}}} \geq 0.5 \cap 0.7 \geq \frac{b_c}{b_{\text{sum}}} \geq 0.3, \quad (11)$$

where  $b_c$  is the result of a white ball counter,  $b_1$  is the result of a number 1 yellow ball counter which is same color as the color of ball number 9, and  $b_{\text{sum}}$  is the sum of all counters. From the former conditional equation, the ball is judged into 4 possibilities such as the cue ball, number 1, number 9, and other balls. The latter conditional equation identifies number 9 ball by the ratio of white. If the ratio of white is over 0.7, the area is white ball. Also, if the ratio of white is less than 0.3, the area is number 1 ball. Otherwise, the area is ball

TABLE 1: Classification by voting method.

Number	c	1	2	3	4	5	6	7	8
(b)	34	55	0	0	0	13	19	1	1
(c)	67	34	0	0	0	2	11	0	8

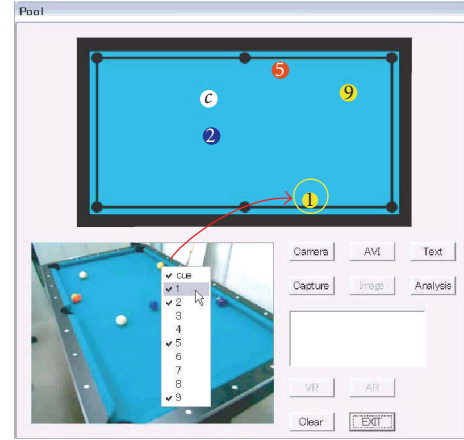


FIGURE 8: Correction by user interaction.

number 9 because the ratio of white of ball number 9 is more than that of ball number 1. These thresholds are empirically determined.

In the case that (11) is not satisfied, a ball number that gets lion's share of votes is the number of the area. In Table 1, (b) is ball number 1 and (c) is ball number 9.

### 3.6. Correction by User Interaction

In the case that some balls stand too close to each other or close to the cushion, the ball areas sometimes cannot be extracted. For instance, the ball area which are close to the cushion that is far side from the camera are not extracted in Figure 8 (the ball with yellow circle). Since the ball area is not extracted, the center of the ball area which is necessary for computing a ball position is not computed. Therefore, the user inputs the center of the ball area and the ball number by using the graphical user interface in our system.

In Figure 8, a user clicks the center 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, our system estimates the ball position by using a projection matrix and displays the ball position on the right pool table image in the interface.

## 4. Computation of Shooting Ways

### 4.1. Rule-Based Strategies

Our system targets NineBall game which is one of popular pool games in the world. The main rule of NineBall game is that players should pocket balls in an order of the ball's numbers. The common rule of pool games is that a player

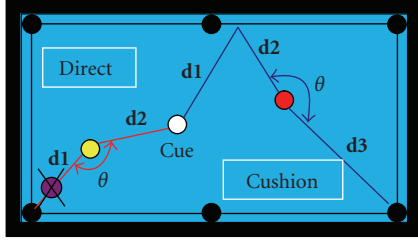
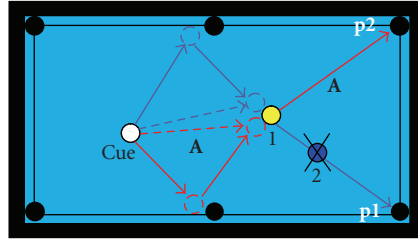
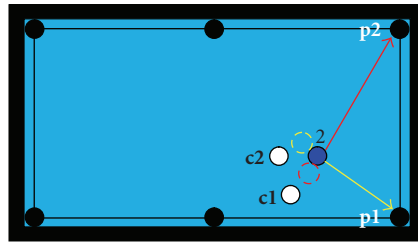


FIGURE 9: Evaluation of ball arrangement.



(a) Ball path



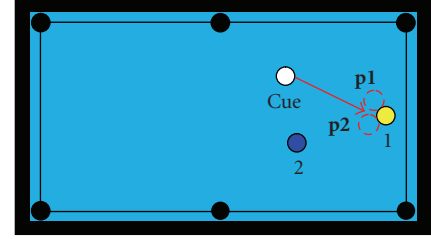
(b) Comparison of two results

FIGURE 10: Pocket.

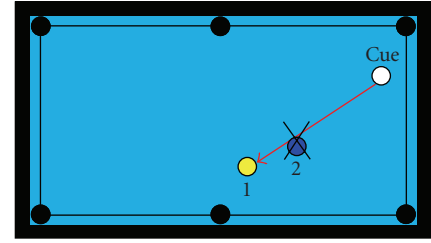
should switch if a player cannot pocket any ball. Based on these two rules, it is necessary to consider shooting ways of Pocket and Safety to win the game.

Pocket means that a player hits the cue ball to pocket balls by considering a position of the next target ball if a target ball can be pocketed, the player can hit the cue ball again. However, a player cannot always hit the cue ball at a target ball because of some special ball arrangement. In the cases, a player targets Safety which means that the player hits the cue ball not to pocket a target ball but to make distribution of the balls complicated to pocket a target ball for another player. This is from the second rule as mentioned above that a current player should switch with the next player if the current player cannot pocket any ball. In NineBall game, it is important for winning the game to judge whether a player can pocket a target ball or not and to consider the shooting way of targeting Safety. For this reason, our system searches some shooting ways of Pocket and Safety considering the next distribution of the balls 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 Free Ball, a player can put the cue ball anywhere.



(a) How to target safety



(b) Result of p1 in (a)

FIGURE 11: Safety.

Thus, our system simulates ball behavior assuming that the cue ball is put on several places.

As for simulation of ball behavior, our system assumes that ball behavior follows particle dynamics and linear motion. Also, our system assumes that a player hits the center of the cue ball.

The algorithm overview is as follows. First, our system searches ball paths of Pocket and Safety for making contact with a target ball directly or using a cushion. Each found path is evaluated by an angle and distances as shown in Figure 9 and as follow:

$$E = a(1 - \cos\theta) + b \prod_{i=1}^n \frac{1}{d_i}, \quad (12)$$

where  $a$  and  $b$  are weighting factors for distances and angles. In a related work [3], these parameters were determined by neural network. Our system empirically determines these parameters as  $a$  on 1.0 and  $b$  on 0.7. Also,  $n$  means the straight paths for targeting the pocket,  $n = 2$  means that the pocket is directly targeted, and  $n = 3$  means that the pocket is targeted by one-time cushion bounds. 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 Figure 9. Next, our system simulates ball behavior by giving several speeds to the cue ball. The distribution of balls in each simulation is evaluated by the same way as shown in Figure 9 and (12). As a result, our system provides some desirable shooting ways.

## 4.2. Pocket

Pocket means that a player hits the cue ball for pocketing a target ball by considering a position of a next target ball. Figure 10(a) represents the ways of making contact with

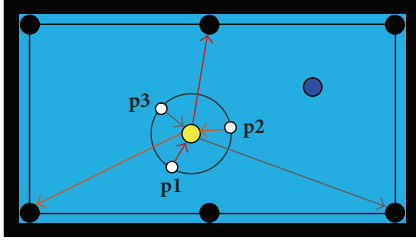
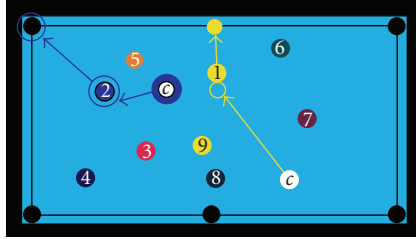
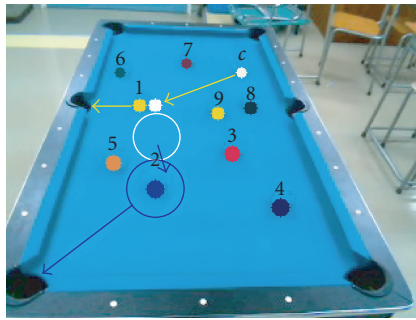


FIGURE 12: Free Ball.



(a) 2D display



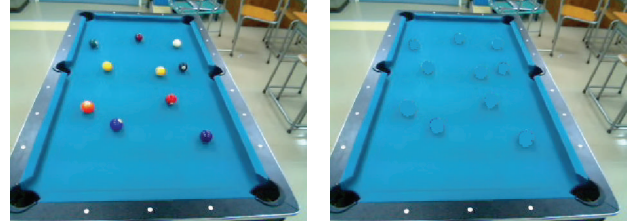
(b) AR display

FIGURE 13: AR display.

a target ball directly and using cushion. Since our system assumes that ball behavior follows particle dynamics and linear motion and a player hits the center of the cue ball, imaginary points of collision with a cushion and a ball that are dot circles in Figure 10(a) are determined uniquely.

In Figure 10(a), there are two paths of each targeting. Since the paths of pocketing ball number 1 to  $P_1$  are hampered by ball number 2, these two paths are reduced. As for two paths of pocketing ball number 1 to  $P_2$ , the ideal point that the cue ball should be stopped after the shot is searched by simulating ball behavior.

As for simulation of path A in Figure 10(a), our system gives the several speed to the cue ball. Figure 10(b) represents two results of the simulation.  $c_1$  and  $c_2$  represents the points where the cue ball stopped in each simulation. As for  $c_2$ , it is difficult to pocket ball number 2 to  $P_1$  because the angle is close to 90 degree. Compared with  $c_2$ ,  $c_1$  is better because the ball distribution is straight. In this case,  $c_1$  is a desirable point. Our system performs this analysis on each ball path in a similar way and computes the desirable point that the cue ball should be stopped in each ball path.



(a) Input

(b) Removal of balls

FIGURE 14: Background.

### 4.3. Safety

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

In Figure 11(a), it is difficult to pocket ball number 1. For this ball arrangement, a player should target Safety. In Safety, collision point with a target is freely determined such as  $p_1$  and  $p_2$  of Figure 11(a). Figure 11(b) represents the ideal result of Safety. In the case of this ball arrangement, next player cannot target number 1 ball directly. Thus, ball arrangement whose score is zero as mentioned above is a desirable Safety. In Safety, our system searches the case that the average of the evaluation of (12) for each pocket is lowest.

### 4.4. Free Ball

Free Ball is an event such that a player can put the cue ball anywhere when an adversary player fouls. In Free Ball, the simulation algorithm is the same as Pocket's algorithm assuming that the cue ball is put on the circumference of a target ball such as  $p_1$ ,  $p_2$ , and  $p_3$  of Figure 12.

## 5. Presentation of Visual Aids

Our system provides visual aids of each shooting way by calibrating the camera. Figure 13(a) represents 2D visual aids that are watched above the pool table. Figure 13(b) is a kind of an augmented reality system in which 3D visual aids such as balls and arrows are drawn on the images that a user is capturing, which is called AR display.

For the AR display, a background image is generated from a captured image by replacing ball areas with the color of the green mat as shown in Figure 14.

Visual aids in AR display are drawn based on online camera calibration by the same way as Section 3.2. The coordinate system of supporting information in Section 4 is the same as Section 3.2. For this reason, the world coordinate system of AR display should correspond to the coordinate system of Section 3.2.

In Section 3.2, left below corner is defined as the origin. On the other hand, each corner has a possibility to be the origin because the image is captured from an arbitrary viewpoint. To solve this problem, our system utilizes the estimated ball positions. First, ball positions are estimated assuming that left below corner corresponds to the origin



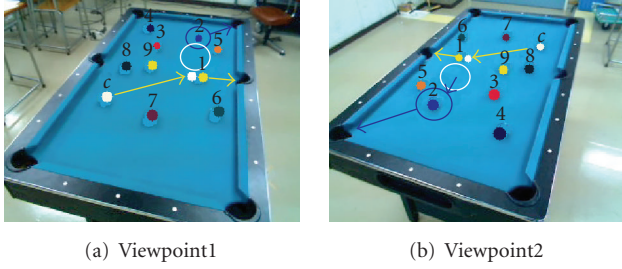


FIGURE 15: Online AR display.

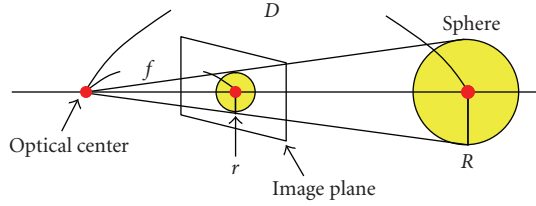


FIGURE 16: Estimation of a ball radius in an image.

which set in Section 3.2. If left below corner is not the origin, ball positions are not equal to the estimated ball positions in Section 3. In a similar way, our system tries estimating ball positions assuming that each corner corresponds to the origin. Our system searched the case that ball positions are equal to the estimated ball positions in Section 3.

Figure 15 represents that a user can see visual aids moving around the table.

As for drawing a ball, our 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 Figure 16 and by the following equation:

$$r = \frac{fR}{D}. \quad (13)$$

## 6. Experimental Results

### 6.1. Ball Position Estimation

For evaluating the accuracy of ball positions, the positions measured beforehand are compared with the results of our system. The size of the pool table is 1330 mm × 700 mm which is smaller than the pool table that is used in official games, the radius of a ball is 24 mm and the size of a captured image is 320 × 240 pixels. Development settings are Pentium M(1.0 GHz) CPU and 512 MB RAM capacity.

Figure 17 represents images that captured from different viewpoints. Table 2 represents the comparison between positions measured beforehand and positions estimated by our system. The maximum error is 16 mm, and the average of errors is 10 mm. We consider that this error is mostly caused by the following some reasons. First, our system assumes that a principal point is approximated to the center of a



FIGURE 17: Accuracy of ball positions.

TABLE 2: Estimated results (mm).

Number	c		1	
Coordinate	$x$	$y$	$x$	$y$
Ground truth	340	404	172	685
View1	340	411	183	691
View2	338	408	173	692
View3	333	398	159	681
View4	352	394	182	678
Number	2		3	
Coordinate	$x$	$y$	$x$	$y$
Ground truth	389	956	524	815
View1	390	954	531	811
View2	399	948	531	806
View3	385	954	533	817
View4	405	959	531	824

captured image. Second, the center coordinate of a ball area is not center because the ball area includes not only the ball color area but also the border with cushion. The corner positions of the table also have a possibility of including error. The average error is close to 1% of the short-side length of the pool table which is 700 mm. The accuracy is sufficient to provide appropriate supporting information to the users because the error did not influence the computation of supporting information as we indicate in next section.

### 6.2. Influence of the Error

In the previous section, we evaluated that the average error in estimating ball position by the proposed method was about 10 mm. We believe that this amount of error is sufficiently small to provide the user the proper strategy of shooting the cue ball. For testing how this error affects computation of the recommended strategy for the user, we performed

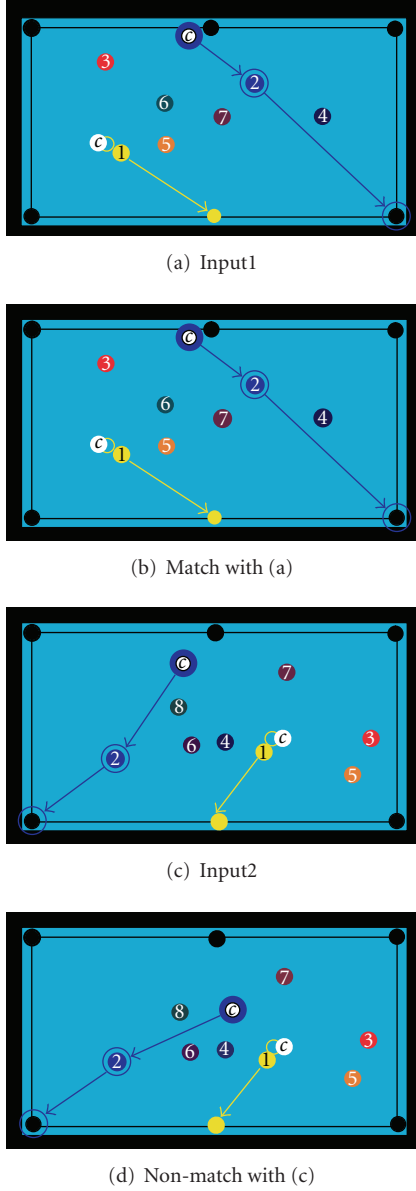


FIGURE 18: Influence of the error.

computer simulation of computing the strategy based on the ball distribution with 10 mm error for each ball position.

In this simulation, we randomly generate a ball distribution, and then compute the strategy for the generated distribution. We also give 10 mm error with random direction to each ball of the ball distribution, and then also compute the strategy for the distribution with 10 mm error. If the computed strategy for the original distribution is the same as the distribution with 10 mm error as shown in Figure 18(a) and 18(b), we can consider that the error does not affect computation of the strategy. On the other hand, we consider that the error influences to computing the strategy as shown in Figure 18(c) and 18(d).

We tested 50 different ball distributions in Free Ball situation. In those 50 distributions, only two distributions

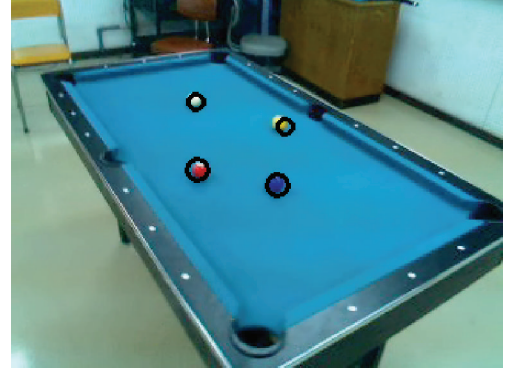


FIGURE 19: Accuracy of AR Display.

TABLE 3: Subjective evaluation of free ball.

Arrangement	User (X)	Our System (Y)	Same
A1	0	3	12
A2	3	9	3
A3	0	3	12
A4	2	10	3

TABLE 4: Subjective evaluation of Figure 20(d).

User	Shot1	Shot2	Shot3
2	1	9(2)	3(1)

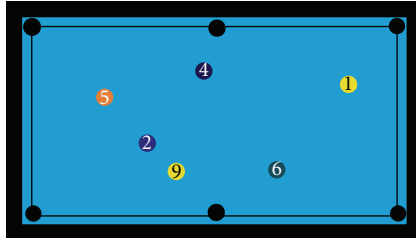
give different strategies for the original distribution and the distribution with 10 mm error. This means that 10 mm error does not give significant influence to compute the strategy from the estimated ball distribution, and it is sufficiently small for the purpose of this paper.

### 6.3. AR Display

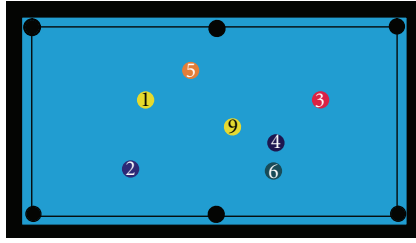
Since the frame rate of AR display is 10 fps, users can see naturally ball behavior drawn on the captured images. For evaluating the accuracy of AR display, the error between projected ball positions and ball positions in the image is computed. In Figure 19, ball positions which are estimated in Figure 17(a) are reprojected on Figure 17(d). The error of the center coordinates between the projected ball and the ball in the image is within 5 pixels on an average. The estimated ball positions include errors as mentioned in Section 6.1. Then, the estimated projection matrix for reprojecting the ball position also includes error. The error of the center coordinates comes from these two error. However, users said that the behavior of the ball which is being pocketed was natural and they did not care about a little error.

### 6.4. Effectiveness of Supporting Information

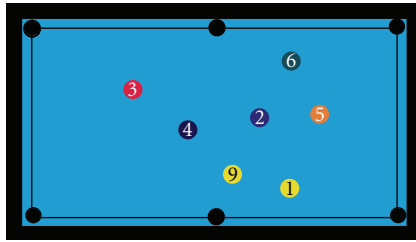
For demonstrating the effectiveness of supporting information, we performed two experiments of subjective evaluation for fifteen beginners who play pool a few times in a year.



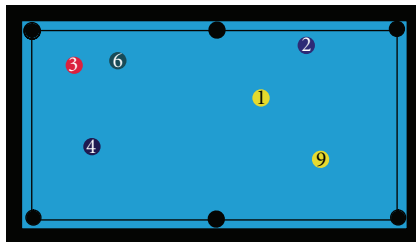
(a) A1



(b) A2

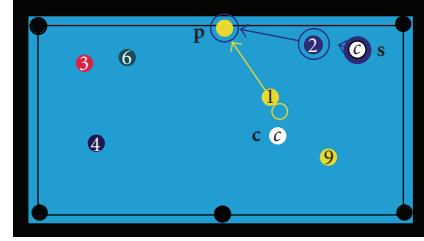


(c) A3

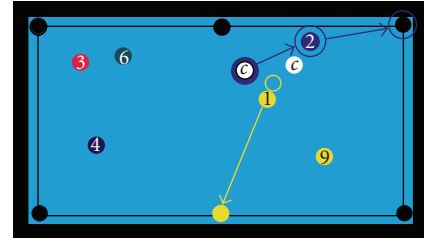


(d) A4

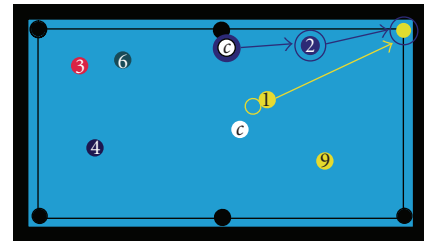
FIGURE 20: Ball arrangement for free ball.



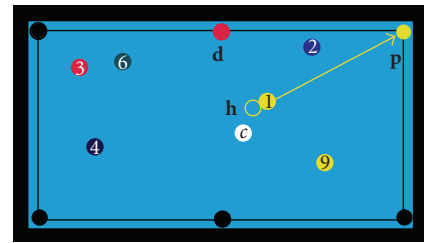
(a) Shot1



(b) Shot2



(c) Shot3



(d) Caution

FIGURE 21: Supporting information of Figure 20(d).

First, an experiment of Free Ball as mentioned in Section 4.4 was performed as follows.

- (1) A ball arrangement is given to the examinees.
- (2) They decide one desirable shooting way (X).
- (3) They watch some shooting ways computed by our system (Y).
- (4) They select one desirable shooting way from (X) and (Y).

Following sentences are explained by using (X) and (Y).

In Table 3, the term “Arrangement” corresponds to the items of Figure 20, the term “User” represents the number of people who selected their consideration (X), the term

“Our System” represents the number of people who selected one of shooting ways computed by our system (Y), and the term “Same” represents the number of people who judged (X) is the same as one of (Y). By counting the number of people, we would like to demonstrate that the shooting ways recommended by our system are actually desirable for the players. This experiment is applied for four kinds of ball arrangement as shown in Figure 20. For each case, the results are shown in Table 3. In the case of (a) and (c), twelve examinees judged that their shooting ways (X) were close to one of the shooting ways of our system (Y). On the other hand, nine examinees in (b) and ten examinees in (d) suggested that one of shooting ways of our system (Y) was more desirable than their shooting way (X). From this table, our system provides shooting ways that are close to the way of

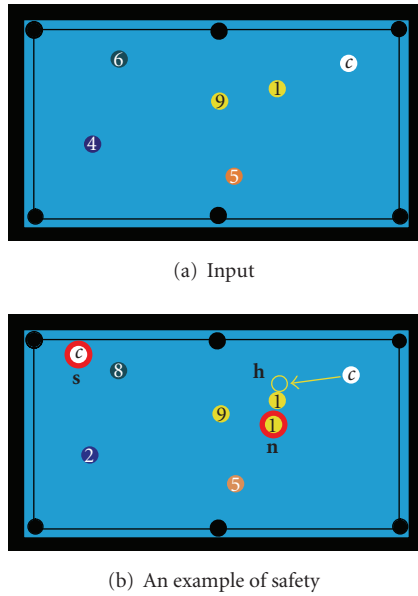


FIGURE 22: Ball arrangement for Safety.

beginners in the case of simple ball arrangement. Our system can also provide more effective shooting ways than the way of beginners in the case that a shooting way is not simply found.

Figure 21 represents the supporting information of Figure 20(d). In Figure 21(a), **c** is the point that the cue ball should be put, **p** is the pocket that ball number 1 should be pocketed, and **s** is the point that the cue ball should be stopped for the next shot. In Table 4, each value represents the number of people who selected one of shooting ways as shown in Figure 21. Nine examinees selected Shot2 and seven examinees did not notice this way first. They said that ball number 1 was easily pocketed and ball number 2 was pocketed twice in a row by Shot2.

In addition, our system can provide information of the caution that the cue ball will be pocketed as shown in Figure 21(d). However, **h** is the collision point of the cue ball and ball number 1, **p** is the pocket that ball number 1 should be pocketed, and **d** is the pocket that the cue ball will be put in. Our system can provide this caution because of simulation of ball behavior. In this ball arrangement that is similar with Figure 21(d), there is a possibility that the cue ball is pocketed if ball number 1 is put into **p**. For this reason, Shot3 includes the possibility that the cue ball is pocketed. Every examinee suggested that this information was very helpful because they did not notice whether the cue ball was pocketed or not.

Second experiment is a detection of a shooting ways of Safety as mentioned in Section 4.3. In this experiment, fifteen examinees watched Figure 22(a) and answered whether they could find a shooting way or not. As a result, twelve examinees could not find the way of hitting the cue ball. They commented that they knew Safety but did not know how to do that and learned the way of Safety from our system. Figure 22(b) represents an example of Safety. In fact, **h** is the collision point with ball number 1, **s** is the point that the

cue ball should be stopped, and **n** is the point at which ball number 1 should be stopped. It is hard to pocket ball number 1 directly in this distribution of balls and this is a desirable Safety.

## 7. Conclusions

We have proposed a system for supporting pool game based on analysis of ball positions for NineBall Game. Our system is implemented on a camera-mounted handheld display and does not use special equipment such as a magnetic sensor and artificial markers. A user can capture a pool table from an arbitrary viewpoint and see supporting information and ball behavior which are drawn on the captured images through the display. In our system, ball positions are estimated by computing a projection matrix of the captured image from an arbitrary viewpoint. As for supporting information, our system simulates ball behavior by giving several speeds to the cue ball and provides some desirable shooting ways based on the rule of the pool game. Moreover, supporting information and ball behavior are online drawn on the images while a user is capturing the pool table. In the experimental results, the accuracy of estimated ball positions is enough for analysis of ball arrangement. In addition, fifteen users evaluated supporting information of our system, and the effectiveness of our supporting information was presented.

## 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.

## References

- [1] M. E. Alian, S. B. Shouraki, M. T. M. Shalmani, P. Karimian, and P. Sabzmejdani, "Roboshark: a gantry pool player robot," in *Proceedings of the 35th International Symposium on Robotics (ISR '03)*, Paris, France, March 2004.
- [2] Z. M. Lin, J. S. Yang, and C. Y. Yang, "Grey decision-making for a billiard robot," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics (SMC '04)*, vol. 6, pp. 5350–5355, The Hague, Netherlands, October 2004.
- [3] B.-R. Cheng, J.-T. Li, and J.-S. Yang, "Design of the neural-fuzzy compensator for a billiard robot," in *Proceeding of the IEEE International Conference on Networking, Sensing and Control (ICNSC '04)*, vol. 2, pp. 909–913, Taipei, Taiwan, March 2004.
- [4] S. C. Chua, E. K. Wong, and V. C. Koo, "Intelligent pool decision system using zero-order sugeno fuzzy system," *Journal of Intelligent and Robotic Systems*, vol. 44, no. 2, pp. 161–186, 2005.
- [5] S. C. Chua, E. K. Wong, and V. C. Koo, "Performance evaluation of fuzzy-based decision system for pool," *Applied Soft Computing*, vol. 7, no. 1, pp. 411–424, 2007.
- [6] T. Jebara, C. Eyster, J. Weaver, T. Starner, and A. Pentland, "Stochasticks: augmenting the billiards experience with probabilistic vision and wearable computers," in *Proceedings*



of the 1st International Symposium on Wearable Computers (ISWC '97), pp. 138–145, Cambridge, Mass, USA, October 1997.

- [7] L. B. Larsen, R. B. Jensen, K. L. Jensen, and S. Larsen, “Development of an automatic pool trainer,” in *Proceedings of the ACM SIGCHI International Conference on Advances in Computer Entertainment Technology (ACE '05)*, pp. 83–87, Valencia, Spain, June 2005.
- [8] H. Uchiyama and H. Saito, “Position estimation of solid balls from handy camera for pool supporting system,” in *Proceedings of the 1st Pacific Rim Symposium on Image and Video Technology (PSIVT '06)*, vol. 4319 of *Lecture Notes in Computer Science*, pp. 393–402, Springer, Hsinchu, Taiwan, December 2006.
- [9] H. Uchiyama and H. Saito, “AR display of visual aids for supporting pool games by online markerless tracking,” in *Proceedings of the 17th International Conference on Artificial Reality and Telexistence (ICAT '07)*, pp. 172–179, Esbjerg, Denmark, November 2007.
- [10] G. Simon and M.-O. Berger, “Reconstructing while registering: a novel approach for markerless augmented reality,” in *Proceedings of the International Symposium on Mixed and Augmented Reality (ISMAR '02)*, pp. 285–293, Darmstadt, Germany, September–October 2002.
- [11] Y. Uematsu and H. Saito, “Vision-based registration for augmented reality with integration of arbitrary multiple planes,” in *Proceedings of the 13th International Conference on Image Analysis and Processing (ICIAP '05)*, pp. 155–162, Cagliari, Italy, September 2005.
- [12] C. Galamhos, J. Matas, and J. Kittler, “Progressive probabilistic Hough transform for line detection,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '99)*, vol. 1, pp. 554–560, Fort Collins, Colo, USA, June 1999.
- [13] Intel, <http://www.intel.com/>.

