私立東海大學

資訊工程與科學研究所

碩士論文

指導教授：石志雄 博士

整合視覺追蹤平台、視覺導引和錯誤分析模型之高效率撞球訓練系統

Integration of a vision-based tracking platform, visual instruction, and error analysis models for an efficient billiard training system

研究生：劉棠鯤

中華民國 九十八年七月

整合視覺追蹤平台、虛擬引導、和錯誤分析模型

之高效率的撞球訓練系統

Integration of a vision-based tracking platform, visual instruction, and error analysis models for an efficient billiard training system

研究生：劉棠緄　　Student：Tang-kun Liu

指導教授：石志雄　　Advisor：Chihhsiong Shih

私立東海大學

資訊工程與科學研究所

碩士論文

Computer Science & Information Engineering

July 2009

Taichung, Taiwan, Republic of China

中華民國九十八年七月

摘要

本撞球軌跡系統結合了視覺導引介面，用來指示使用者做出可靠的擊球。整合的系統平台是以個人電腦(PC)運行。本視覺系統可用在母球、子球和球桿的軌跡偵測。使用最小平方差法將現實世界跟虛擬世界的撞球座標做校正，以計算出精確的導引線。使用者能夠根據個人電腦螢幕上所顯示的視覺導引線，在撞球桌上調整擊球桿的位置。

根據碰撞移動分析來計算出母球理想的視覺導引線，除了要計算出理想的視覺導引，在不同的子球跟袋口中選擇最佳擊球點，也是一個須要探究的影響因素。由子球與袋口的理想導引線，其可容忍進袋的角度大小就代表一個成功擊球的難度，而難度是依序地根據袋口與子球的距離、子球與母球的距離、和這兩個向量的角度大小，此三者來決定。

將這些功能條件來模擬容忍角度的測試。選定一子球廣泛的使用不同的幾何參數做測試，包括使用及不使用本整合系統所得到的結果。並選擇不同熟練程度的球員來做實驗。結果顯示所有的球員在技巧的加強方面，都因為我們提出的視覺導引系統而得到了改善，而技術較差的球員更因為本系統的幫助而改善了最多的技巧。包括最大和平均擊球率，所有得到的結果都獲得了改善。擊球率的實驗結果範例跟我們的分析是一致的。擊球率和分析出的子球入袋的容忍角度有著緊密的關聯。種種結果證明了本系統良好的效能，並且此分析結果可以用於有效的遊戲競賽策略。

關鍵字：視覺追蹤，擊球誤差分析，系統整合，最小平方差校正，擴增實境

Abstract

A billiard ball tracking system is designed to combine with a visual guide interface to instruct users for a reliable strike. The system makes use of a vision system for cue ball, object ball and cue stick tracking. Users are able to adjust the cue stick on the pool table according to a visual guidance line instruction displayed on a PC monitor. In addition to calculating the ideal visual guide, the factors influencing selection of the best shot among different object balls and pockets are explored. It is found that a tolerance angle around the ideal line for the object ball to roll into a pocket determines the difficulty of a strike. This angle depends in turn on the distance from the pocket to the object, the distance from the object to the cue ball, and the angle between these two vectors.

Simulation results for tolerance angles as a function of these quantities are given. A selected object ball was tested extensively with respect to various geometrical parameters with and without using our integrated system. Players with different proficiency levels were selected for the experiment. The results indicate that all players benefit from our proposed visual guidance system in enhancing their skills, while low-skill players show the maximum enhancement in skill with the help of our system. All exhibit enhanced maximum and average hit-in rates. Experimental results on hit-in rates have shown a pattern consistent with that of the analysis. The hit-in rate is thus tightly connected with the analyzed tolerance angles for sinking object balls into a target pocket. These results prove the efficiency of our system, and the analysis results can be used to attain an efficient game-playing strategy.

Subject terms: vision tracking; strike error analysis; system integration; least squares error calibration; augmented reality.

Table of Contents

List of Tables

List of Figures

Chapter 1 Introduction

1.1 Motivation

Billiards is a popular game, which has been played for hundreds of years in some form. The game is played by striking a cue ball with a cue stick, causing the cue ball to collide with another ball the object ball so as to drive the object ball into a selected pocket on the playing surface. To achieve an acceptable level of proficiency in the game requires considerable practice. Because this can be frustrating and unfruitful for beginners, and at times even for more advanced players, numerous learning aids have been devised over the years to assist players in developing and enhancing their proficiency both in the real game and in virtual environments.

In Table 1, the techniques used by this paper are compared with other relevant work, and their uniqueness is bolded. We can observe that this paper applies an error analysis model to evaluate the chances of a successful hit. Also, the cue stick was not tracked by the other techniques. By tracking the cue stick, the user can have better control of the strike, given a theoretically analyzed guidance line. The way user learns about the hit stroke is also unique. The visual display of the instruction results on a PC monitor helps to reduce the cost of the system.

Table 1 Comparison of pool learning tools and setups.

	Vision tracking		Motion analysis strategy		Service target		Visual guidance instruction media			
Work	Ball center	Cue stick	Fuzzy logic	Tol. error model	Robot	Human	PC monitor	LCD eye goggle	Laser light pointer	Hardware cost
Chua et al.	No	**No**	Yes	**No**	No	No	**No**	No	No	**Low**
Chua et al.	Yes	**No**	No	**No**	Yes	No	**No**	No	No	**High**
Jebara et al.	Yes	**No**	No	**Yes**	No	Yes	**No**	Yes	No	**High**
Cheng et al.	Yes	**No**	Yes	**No**	Yes	No	**No**	No	No	**High**
Larsen et al.	Yes	**No**	No	**No**	No	Yes	**No**	No	Yes	**High**
Nakama et al.	Yes	**No**	No	**No**	Yes	No	**No**	No	No	**High**
This thesis	Yes	**Yes**	No	**Yes**	No	Yes	**Yes**	No	No	**Low**

Note: Bold indicates the uniqueness of the author's system

1.2 The Goal and Contributions

In this work, a novel vision-based billiard ball tracking system is designed to provide the player an interactive guiding system to orient the cue stick properly on the pool table. The major goal is to increase the aiming accuracy during the hitting process to help increase the fun of playing this game without a complex electronic or mechanical setup on or around the playing table. The system achieves the goal by tracking the actual ball's center position and the cue stick orientations on the table using a real-time vision system. A graphical user interface is provided to display the actual balls and cue stick positions derived from the vision system. A theoretically calculated ideal collision line is drawn from the cue ball on the display system, given the objective ball and target pocket locations.

The target pocket and object ball are selected in the visual display system, based on a motion analysis algorithm for a best shot. An imaginary guidance line is calculated from the basic physical laws of collision and will change its orientation around the cue ball according to selected target pocket and object ball positions. The user then moves the cue stick on the pool table, which is traced by the vision system. The cue stick centerline is represented by another imaginary line in the display system. The user then adjusts this centerline by moving the cue stick around to match the calculated ideal collision line from the cue ball. Once the two lines align with each other in the visual display, the user can strike the cue ball and watch the object ball roll into the selected pocket.

The ideal collision line from the cue ball is derived from motion analysis. The analysis has shown that the tolerance angle around a nominal aiming direction determines the success rate for sinking an object ball into a pocket and is a function of the cross angle between pocket, object ball, and cue ball, the distance from pocket to object ball, and the distance from cue ball to object ball. An analytical solution is derived for the tolerance angle. Simulation results are then provided by changing these geometry parameters.

1.2 Thesis Organization

The article first presents the system setup and the visual interface, followed by image calibration to accurately correlate the ball center pixel coordinates with the actual positions on the billiard table. Then, the algorithms developed in processing and recognizing the images during the tracking of the cue stick, the cue, and the object balls is discussed. An analysis is performed to calculate the tolerance angle around a nominal ideal guidance line. The tolerance angle gives a measure of how hard it is to sink an object ball. The factors influencing this quantity are explored. A theoretical model is built to understand the interrelation between them. Finally, an extensive test drive of the whole system, including back-end tracking and front-end visual display, is performed to verify the accuracy of the system by changing these parameters. We specify different combinations of the cross angle between pocket, object ball, and cue ball and the distance from cue ball to object ball. The sink rate is recorded as a function of these parameters. The experimental results are then compared with the analytical models. Important issues and future work are then presented.

Chapter 2 Background

Chua et al. applied the concept of fuzzy logic to develop the decision algorithm for a best shot in computer-generated pool environments.[1] Recently, researches have been carried out to create intelligent robots in many applications.[2–7] These include robot golf,[2] the yo-yo,[3] volleyball,[4] chess,[5,6] and table tennis.[7]

Some published papers show some of the needed functions for a billiard robot. Chua et al. demonstrated ball identification and calibration for a pool robot by the image-processing technique.[8] Nakama et al. developed a shooting mechanism for a billiard robot utilizing a precise position mechanism.[9] A wearable computer and augmented reality have been demonstrated by Jebara et al.[10] to help players to enhance their game of billiards. A vision algorithm is implemented that operates in interactive time with the user to assist planning and aiming. Probabilistic color models and symmetry operations are used to localize the table, pockets, and balls through a video camera near the user's eye. Classification of the objects of interest is performed, and all possible shots are ranked in order of their usefulness.

Most of the learning aids that have been researched, however, are complex and unwieldy. Many require that some type of gadget or attachment be positioned on or around the actual playing table or on the player.[10–12]

Cheng et al.[12] have designed a billiard robot to imitate the ability of human beings to learn to play billiards. The objective is to design a neural fuzzy compensator for this robot to improve its skill. First a model for predicting the hitting error is developed, based on a recorded database of pocketing processes. Then, the predicted error is compensated by the fuzzy controller to decide the cutting angle hitting point of the object ball automatically.

Larsen et al.[13] describe an automated pool trainer APT , a multimodal pool training system developed at Aalborg University. The idea of the system is to automate the learning process. It utilizes spoken interaction combined with a graphical output and a computer-controlled laser pointer for user communication. The trainee selects a suitable exercise among a number of predefined courses aided by the system . The system issues instructions on how to place the balls on the table, shows the optimal shot, and records and evaluates the performance of the player. The instructions are given orally speech synthesis combined with gestures using a laser as a virtual pointer on the pool table. There is no error analysis or automatic motion selection strategy supporting the system. A human expert is behind the system to plan the training courses. Also, the laser pointer can cause visual harm to personnel under training.

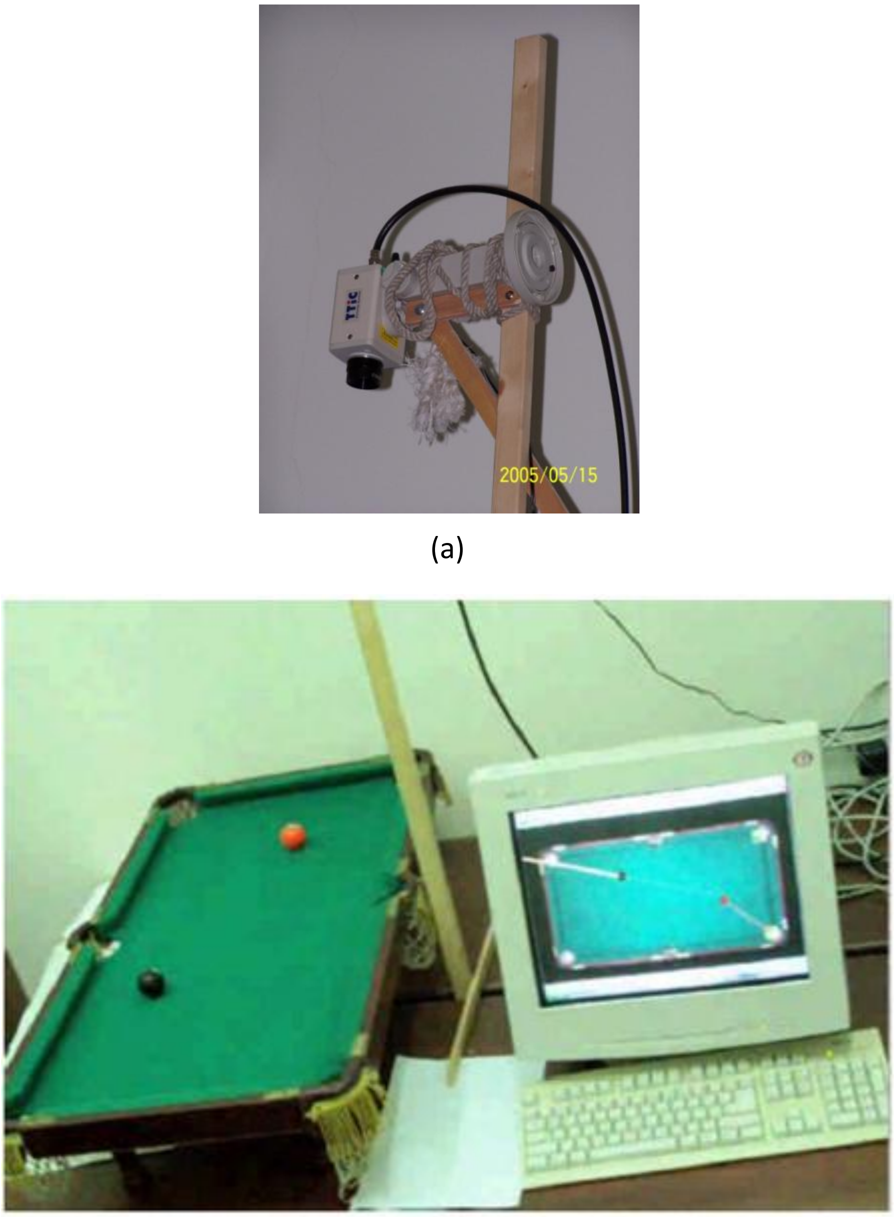
For these reasons, these devices are often not helpful to the learning process. Many are not popular in a family setting, and most are prohibited from commercial playing tables. Table 1 summarizes relevant pool learning tools and setups and compares their pros and cons with those of this paper. The indices of comparison include vision tracking targets, motion analysis strategy used, service target, and visual guidance methods.

Chapter 3 Proposed Approaches

3.1 System Description

The system is composed of a CCD camera, a vision card, a pool table, a cue stick with tagged tip, and a PC running a graphical user interface as shown in Fig. 1. The CCD camera is mounted directly above the billiard table on a set of fixtures. The camera orientation is set arbitrarily; the only requirement is that the field of view must cover the whole billiard table with a minimum amount of surrounding environment pixel information enclosed. The tip of the cue stick is tagged with two stripes of green cloth a small distance away from each other, creating two isolated regions of different color from the surrounding table. A Pentium III PC running an interactive graphical user interface sits right next to the billiard table.

The software in the PC executes both the visual display that instructs the user how to place the cue stick on the pool table, and the processing of the ball and the cue stick movement images as grabbed by the CCD sensor.



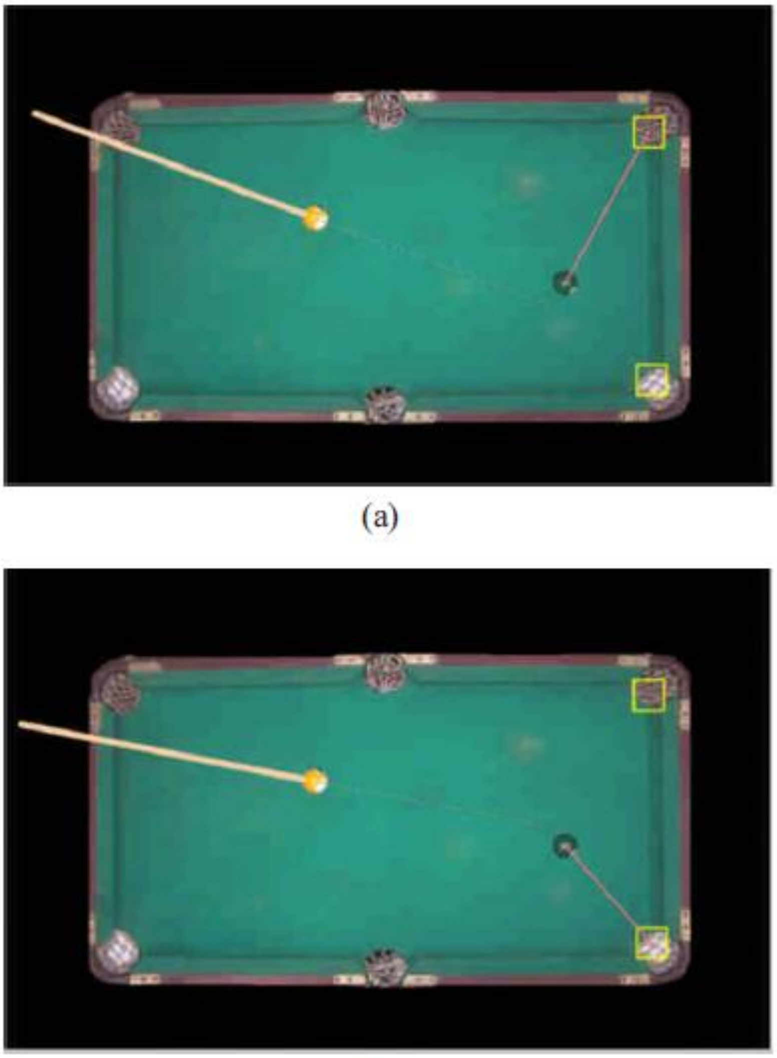
(a)

(b)

Figure 1 System setup and configuration.

3.2 Learning Model and Visual Display

The visual display system plays the role of presenting the user a strategy for driving the cue stick. Given the combination of target pocket and object ball selected by the motion analysis results in Sec. 3, a correct aim direction is drawn from the center of the cue ball as a guide for players to drive the cue stick. This aiming direction is presented as a light green line as shown in Fig. 2. The procedure to derive this guidance line is based on the physical laws of collision. The centers of the target pocket and object ball are first connected and extended beyond the object ball as shown in Fig. 3. The collision point is then selected as a point alone the extension line that is exactly one ball diameter ***d*** away from the object ball center as marked in Fig. 3. Next we connect the collision point with the cue ball center to form an extension line drawn beyond the cue ball center as a light green line in Fig. 2.



(a)

(b)

Figure 2 Visual interface showing calculated guidance line.

The selection of the combination of pocket and object ball will influence the orientation of the guidance line. Two such selections are visualized in Fig. 2(a) and

2(b). The light green line is seen to be pointing in different directions, for the same cue ball and object ball locations, with different target pocket locations.

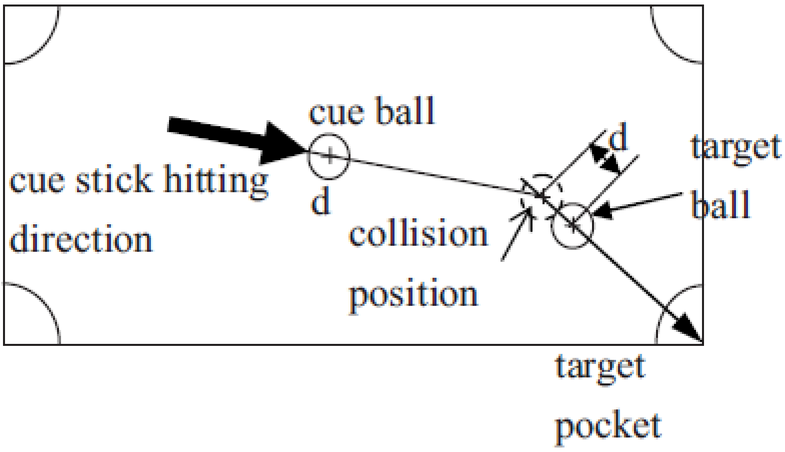
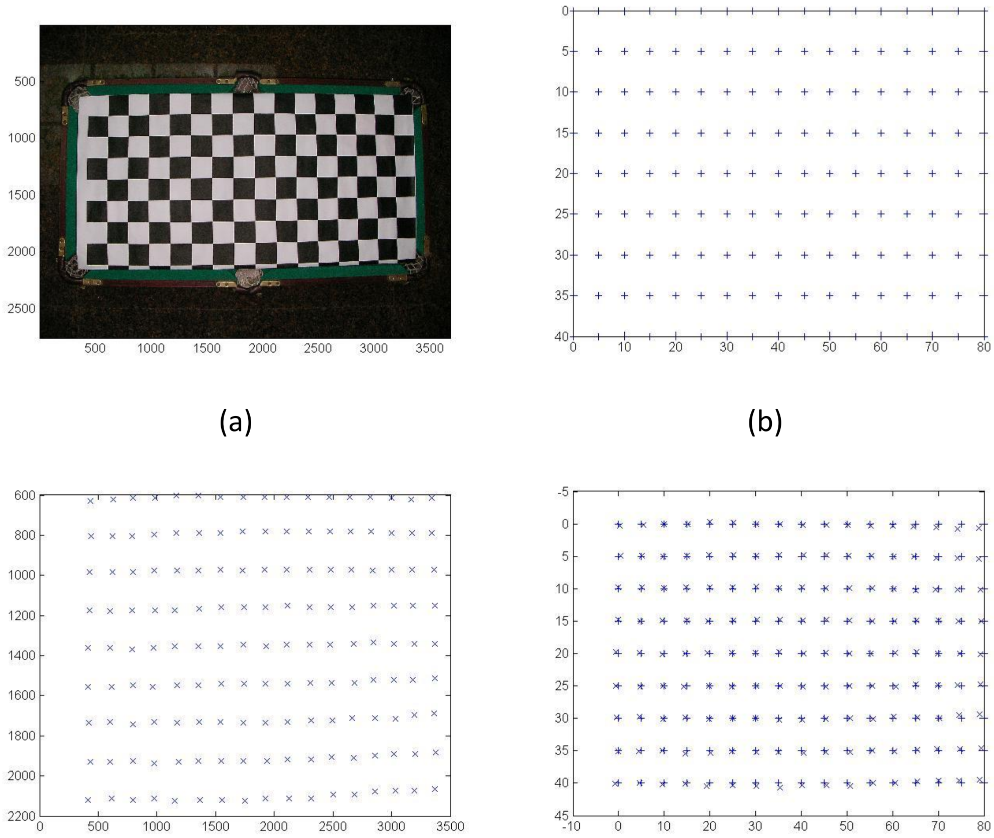


Figure 3 Guidance line model.

3.3 Calibration and Image Processing Algorithm and Their Integration with Visual Display

Before the system can grab pictures and analyze the image, a calibration procedure is undertaken to correlate the image coordinates with the actual coordinates on the billiard table. A calibration board with a grid pattern is printed and placed on the billiard table as shown in Fig. 4(a). The dimensions of this mini pool table are about 40 × 80 cm. The camera is then triggered to take a picture of this calibration board. The actual coordinates of the intersection point of the grid line structure on the calibration board are measured as in Fig. 4(b). These coordinates are then stored in a

$$153 \times 2 \text{ matrix}, A = \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ \vdots & \vdots \\ x_{153} & y\vdots \end{bmatrix},$$



(a) (b)

(b) (d)

Figure 4 Calibration charts and superimposed error map.

where (x_n, y_n), n = 1 , . . . , 153, is the n'th grid point on the calibration board of the billiard table. The coordinates are in centimeters. The image coordinates of the grid intersection points are extracted using a regular thinning and edge extraction image-processing algorithm as in Fig. 4(c). The image coordinates are then stored in a

$$153 \times 3 \text{ matrix,} \quad B = \begin{bmatrix} ix_1 & iy_1 & 1 \\ ix_2 & iy_2 & 1 \\ \vdots & \vdots & \vdots \\ ix_{153} & iy_{153} & 1 \end{bmatrix},$$

where (ix_n, iy_n), n = 1 , . . . , 153, is the n'th grid image pixel position. A correlation matrix, T, is then calculated using a least-squares error transformation equation $T = (B'B)^{-1}(B'A)$.

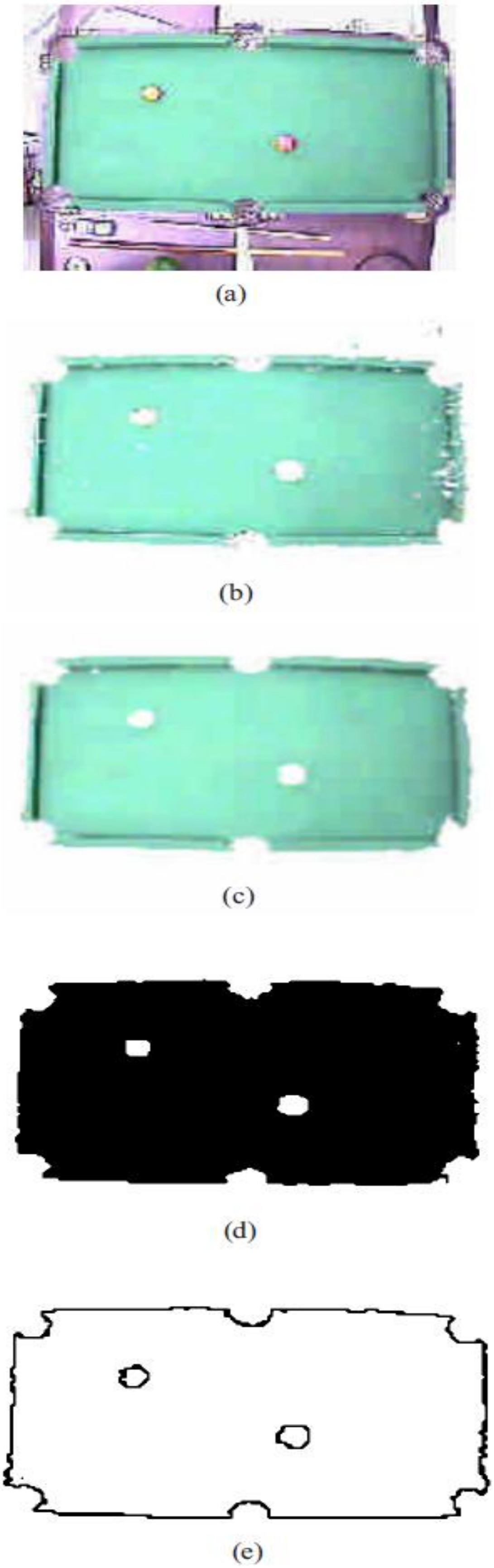
By this transformation matrix (3 × 2) , the image coordinates are transformed back to the real-world coordinates on the pool table via the equation A' = BT, and superimposed on the original coordinate matrix A to form Fig. 4(d) . The standard deviation error between original and transformed coordinates for each grid point is on the order of 0.1 cm.

The image-processing procedure needs first to calculate the centroid positions of the cue ball and the target ball. Figure 5(a) is a raw image of the billiard table including a cue ball and an object ball. An RGB-to-HSV conversion is performed for each pixel as follows:

$$H = \cos^{-1}\left\{\frac{\frac{1}{2}[(R-G)+(R-B)]}{\sqrt{(R-G)^2+(R-B)(G-B)}}\right\}, \qquad (1)$$

$$S = \frac{Max(R,G,B)-Min(R,G,B)}{Max(R,G,B)}, \qquad (2)$$

$$V = \frac{Max(R,G,B)}{255}. \qquad (3)$$



(a)

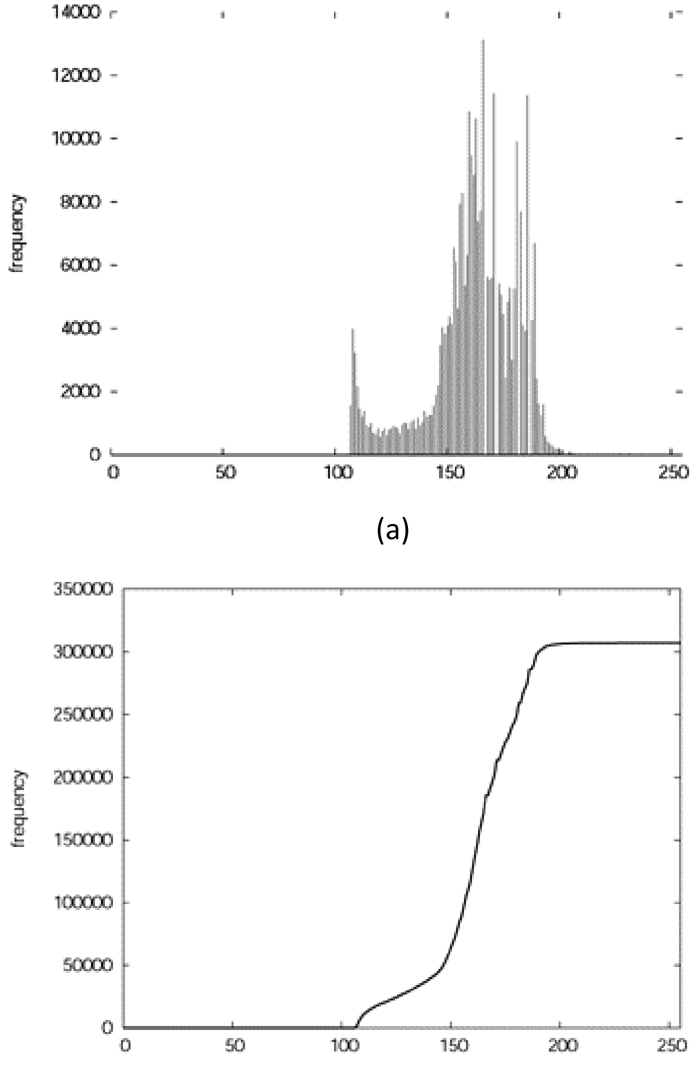
(b)

(c)

(d)

(e)

Figure 5 Boundary edge extraction.



(a)

(b)

Figure 6 Integration data flow

A histogram chart is then built based on the H values for all the pixels of an empty table image in Fig. 6. A range of H values from 110 to 185 is selected to collect the table pixels, since most of those pixels exhibit such values and possess the highest peaks in a histogram of H versus number of pixels from the image of Fig. 5(b). The unwanted pixels, including those of the background and possibly the ball and stick images, can then be filtered out. An initial result is shown in Fig. 5(b). A median filter algorithm is then executed to smooth the images as in Fig. 5(c). A typical thresholding and Sobel operator are used to segment out the boundaries of the table, cue ball, and target balls as in Fig. 5(d) and 5(e) . The boundary pixels of each independent object are further extracted by a recursive traversing process that groups them into different sets. A region scan is then performed inside each of the separated sets to identify the color of each, so as to decide if it belongs to a colored ball or a stick tip. Tracking of the cue stick is accomplished by tracing the two separate colored regions on the tip of the stick.

The back-end vision system and front-end visual display has been fully integrated with this prototype mini pool table as shown in Fig. 15. (Sec. 4) The data flow between the different subsystems is illustrated in Fig. 7. The vision system first grabs and analyzes the cue stick, object and cue ball, and pocket center pixel coordinates. This information is passed to both the visual display system and the coordinate transformation step ***a*** as shown in Fig. 7. Step ***a*** converts the image coordinates into world coordinates on pool table through a transformation matrix in the calibration process. Step ***b*** calculates the theoretical strike line using the world coordinates of the cue and pocket, based on the collision model and analysis results. The derived centerline is then converted back to the visual world using the inverse transformation procedure of step ***c***. Finally, the visual display system can display both the real-world objects and an instruction line, which users can rely on for a precise stroke to sink the object ball into a selected pocket.

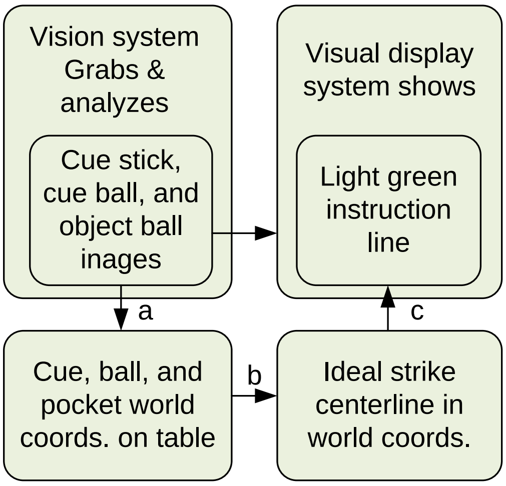
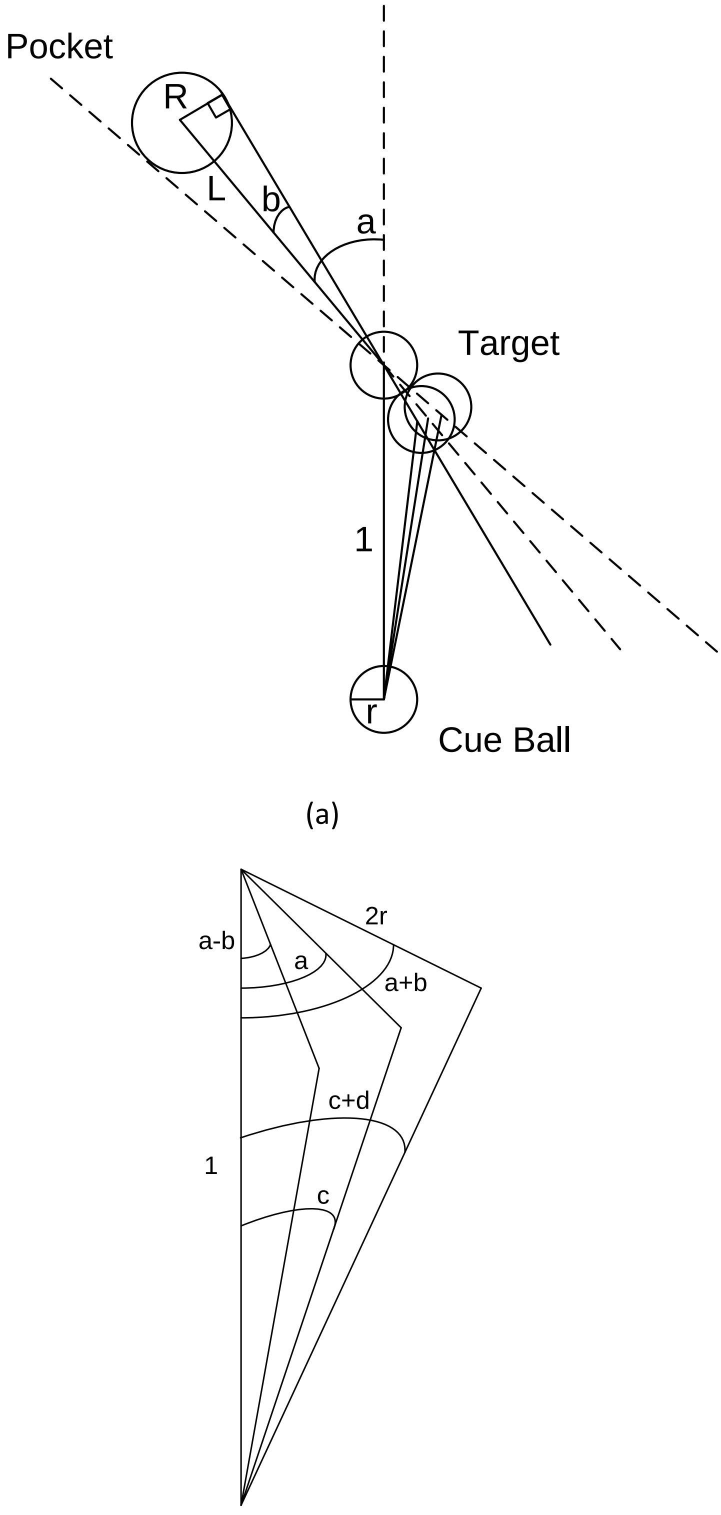

Vision system Grabs & analyzes
Cue stick, cue ball, and object ball images
Visual display system shows
Light green instruction line
a
c
Cue, ball, and pocket world coords. on table
b
Ideal strike centerline in world coords.

Figure 7 Integration data flow.

3.4 Error Analysis Models

Figure 8 shows the results of such calculations. The profiles are intuitively reasonable, with the maximum tolerance at zero angle ***a*** and about one ball diameter distance between the cue and object balls. Generally, this value decreases gradually with increasing ***l*** at constant ***a***. The slightly reduced tolerance angle at close range with the object balls for larger ***a*** has been found to be due to behavior of Eq. (6). We now describe a strategy for selecting the shot that is most likely to succeed. Starting from the cue ball, we compute linear trajectories to all solid balls. We then compute linear trajectories from these to the detected pockets. Then we test each of these possible trajectories, from cue ball to solid ball to pocket, to find those that do not involve any collisions with other objects. If no solid balls are found, we test the ball for possible trajectories.



(a)

(b)

Figure 8 Schematic of cue ball collision tolerance angle definition.

We then weight each of the valid trajectories by the following heuristics. Each time the user is to take a shot, we are really interested in the angle at which he or she hits the cue ball (we assume that the force with which the cue ball is hit is going to be determined by the user and enough to sink the object ball.) So, what is fundamentally being suggested to the user is an angle at which to hit a cue ball, say the angle ***c*** as a deviation from the line connecting the cue ball to the solid object ball. The more accuracy we need on the angle c, the harder the shot. Figure 7 shows how we can determine the required angle, using Eqs. (4)–(6). The distance from the cue ball to the solid ball is ***l***, and the distance from there to the pocket is ***L***. The angle formed by the three objects is the angle ***a*** at the solid ball. We can compute ***c*** using this information, as well as a bound on the maximum error on ***c***, which is denoted by ***d***.

For each possible shot, we compute the angle d and display to the user the shot with the greatest value of d (the shot that requires the least accuracy) with the required trajectory (at the angle c) superimposed and extended from the cue ball on the visual display. This is a simplified first-order strategy model, since it is only 2-D and does not include spin effects, kinematics, or rebounds in the computation of the shots. We have

$$b = \sin^{-1}(R/L) \tag{4}$$

$$c = \sin^{-1}\left(\frac{2r\sin(a)}{\sqrt{4r^2 + l^2 - 4rl\cos(a)}}\right) \tag{5}$$

$$d = \sin^{-1}\left(\frac{2r\sin(a+b)}{\sqrt{4r^2 + l^2 - 4rl\cos(a+b)}}\right) - c \tag{6}$$

Based on this analysis, the collision tolerance angle ***d*** is in fact a function of ***l***, ***L***, and ***a***. The larger the tolerance angle is, the easier for the player to sink the object ball. Given a fixed ***L*** value (a ratio 1/3), this paper first evaluates the tolerance angle value on the different combination of ***l*** and ***a*** values, each ranging from 0 to 50 and from 0 to 90 degree, respectively. A subsequent paper will discuss the effects on the tolerance angles of fixing one parameter while varying the other two. The ***l*** value is in fact a multiple of the radius of the billiard ball (r).

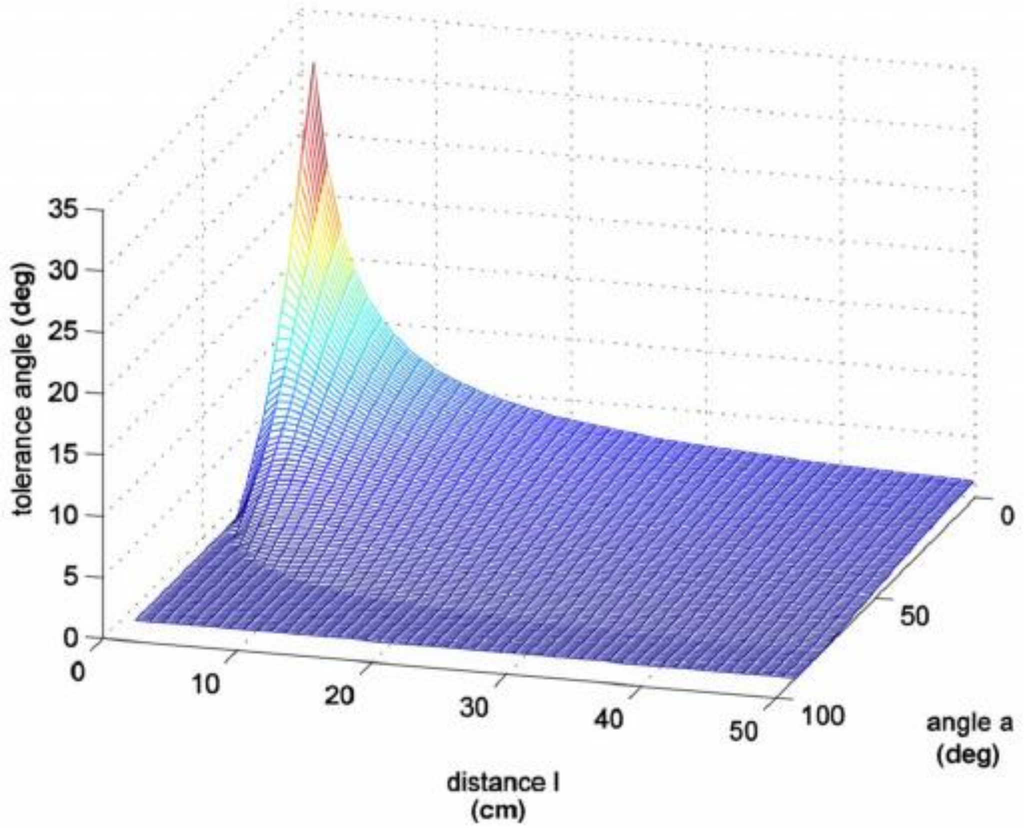


Figure 9 Calculated profile of the cue ball tolerance angle ***d***.

Chapter 4 Experiment Results

4.1 Numerical Results

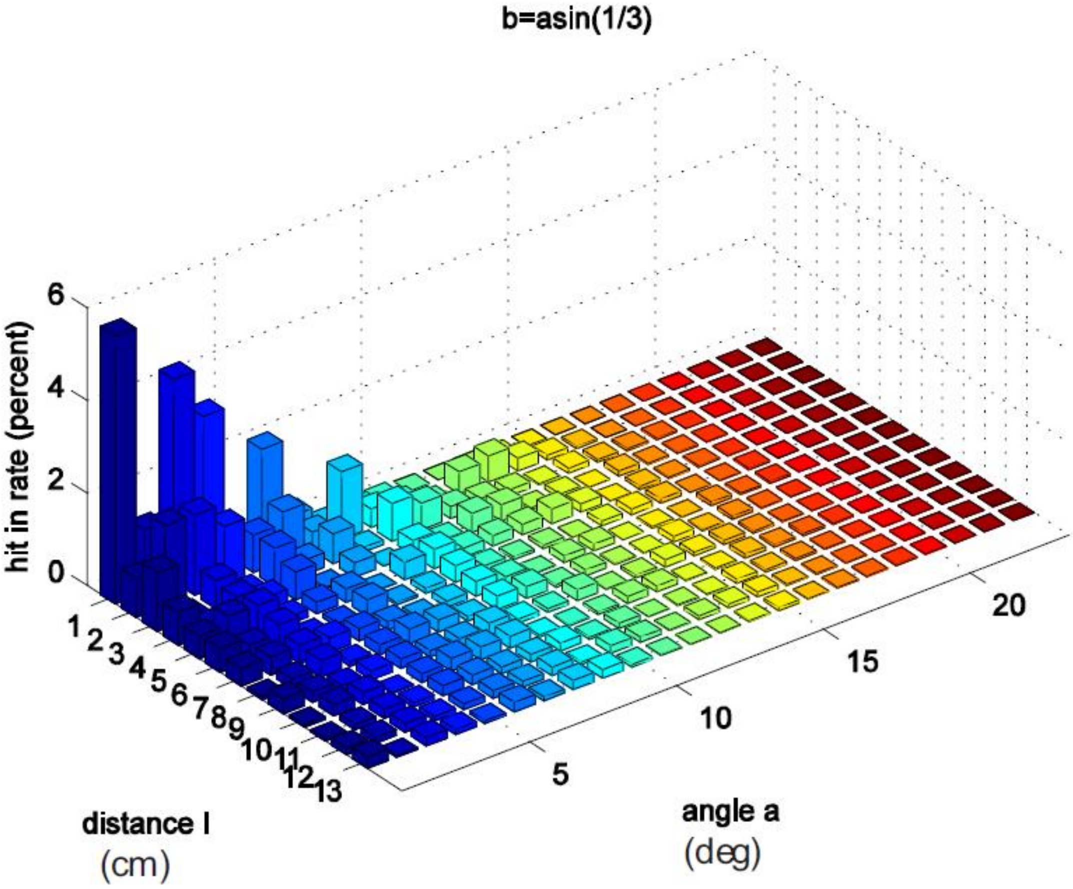


Figure 10 Hit-in rate as function of collision angle ***a*** and distance ***l*** for low-proficiency player without guidance.

We then test the hit-in rate at each combination of cue ball angle and distance between cue and object balls. The three players has been given our pre-test to decide the low, median, high level of proficiency. Three players with different levels of proficiency were chosen to test our system. Firstly, two sets of experiments were conducted for each player. One set did not use any guidance lines, while the other used the guidance line provided by our proposed system for aiming. The object ball is fixed at a distance about three times the pocket radius for both sets. At each combination of cue ball angle (***a***) and distance between cue and object balls (***l***), we calculate the ideal visual guidance line using the tracked ball center points from the vision system. Fifteen shots were played for each combination of cue ball angle and distance between cue and object balls for both sets of experiments for the three players. The success rate is then calculated as the number of sink shots divided by the total number of shots. The results are plotted as a function of $\boldsymbol{l}$ and $\boldsymbol{a}$, for sets of three players playing without guidance lines, in Figs. 9–11 with low, medium, and high proficiency, respectively. The same testing arrangements, but this time with guidance lines added, were then conducted for the same players. The results are again plotted as a function of $\boldsymbol{l}$ and $\boldsymbol{a}$ in Figs. 12–14 for low-, medium-, and high-proficiency players, respectively.

The comparisons of statistics of these two sets of data are further summarized in Tables 2 and 3 for experiments without and with visual guidance, respectively. The indices used for comparison in both tables include the maximum hit-in rate, the average hit-in rate, and a measure of similarity to the profile of the analysis results for the tolerance angle.

The similarity measures are based on the l_2 norm of the experimentally measured data. In other words, a query is measured scored based on an appropriate distance measure in the data measurement space. Suppose the measured two-dimensional hit-in rate and the theoretical tolerance angles are recorded and transformed into one-dimensional vectors $S_A = [a_1 , a_2 , a_3 , . . .]$ and $S_B = [b_1 , b_2 , b_3 , . . .]$, respectively. The size of the one-dimensional vector is equal to the product of the numbers of columns and rows of the two-dimensional matrix. A datum in row $\boldsymbol{i}$ and column $\boldsymbol{j}$ in the two-dimensional matrix is allocated a space in the one-dimensional vector as follows: $\boldsymbol{i}\times$row size+ $\boldsymbol{j}$ + 1. The tolerance angle vector is further normalized to a percentage, the maximum being 100%, by dividing each angle by the maximum value at each combination of $\boldsymbol{l}$ and $\boldsymbol{a}$. The distance measure between two vectors in the data space can be calculated as

$$\|d\|_2 = |S_A - S_B|^2 = \left(\sum_{i=1}^{n} |a_i - b_i|^2 \right)^{1/2} . \qquad (7)$$

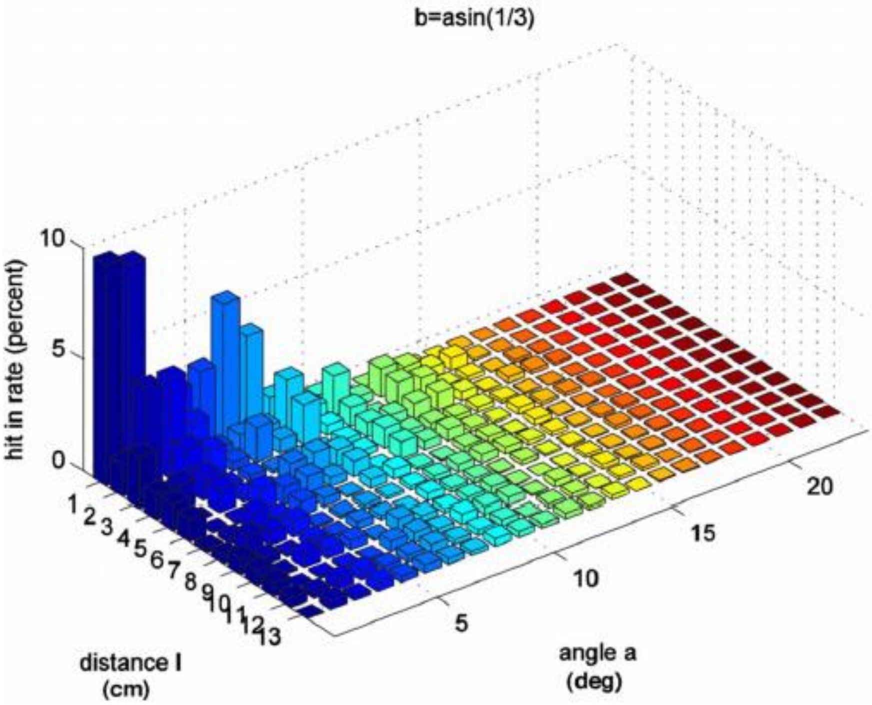


Figure 11 Hit-in rate as function of collision angle ***a*** and distance ***l*** for medium-proficiency player without guidance.

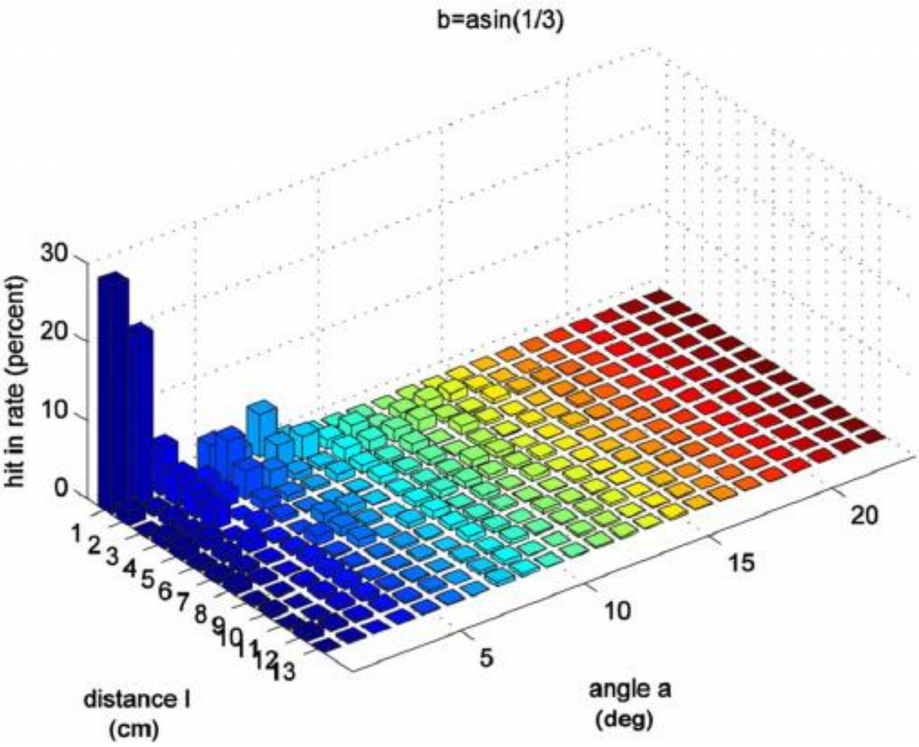


Figure 12 Hit-in rate as function of collision angle ***a*** and distance ***l*** for high-proficiency player without guidance.

The difference between the amplitudes (D_A) can be normalized and rescaled to be between 0 and 100 as follows:

$$D_A = 100\frac{\|d\|_2}{\left[\sum_{i=1}^{n}\max\left(a_i,b_i\right)^2\right]^{1/2}}. \qquad (8)$$

The similarity between two data profiles can then be measured as SA = 100− D_A, so that a score of 100 denotes that the profiles are identical, and 0 that they are the least similar profiles: $S_A[c,c'] = 100 - D_A$ (where $D_A \in [0,100]$). (9)

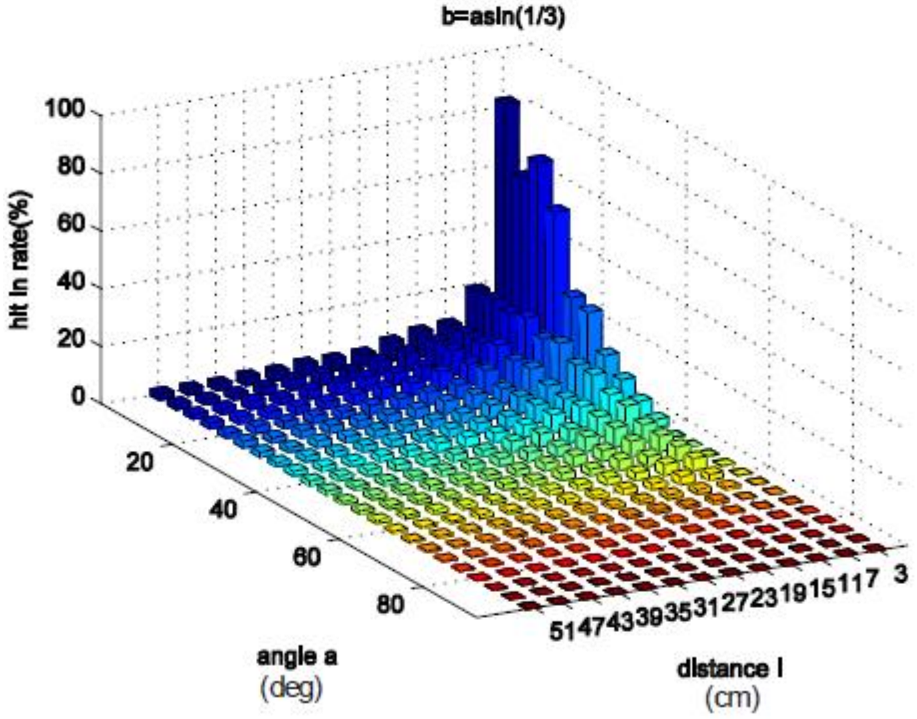


Figure 13 Hit-in rate as function of collision angle ***a*** and distance ***l*** for low-proficiency player with guidance.

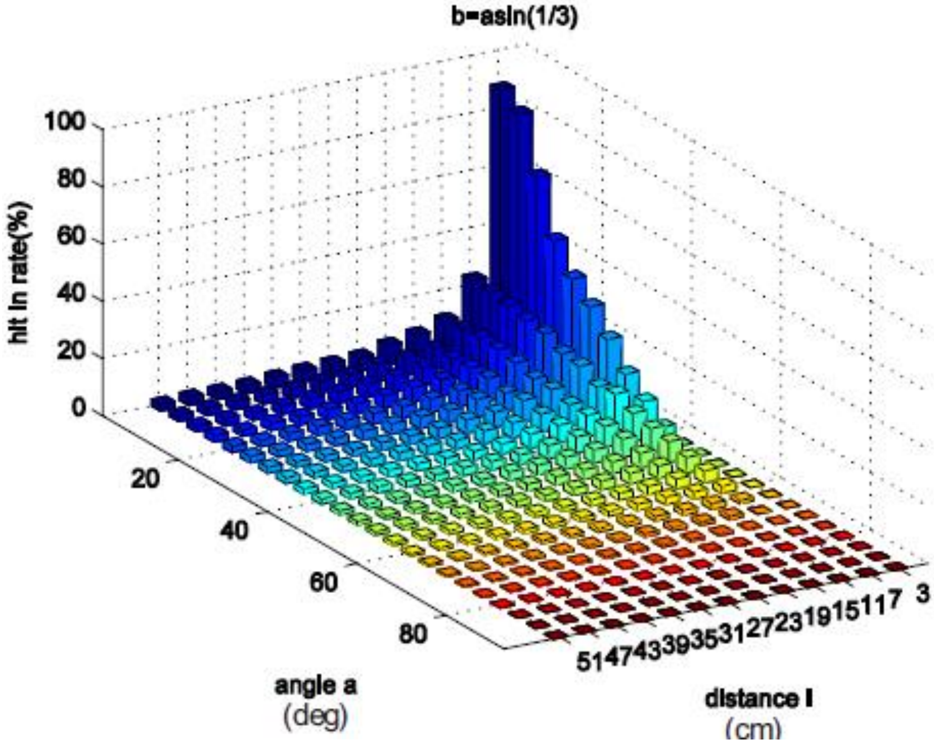


Figure 14 Hit-in rate as function of collision angle ***a*** and distance ***l*** for medium-proficiency player with guidance.

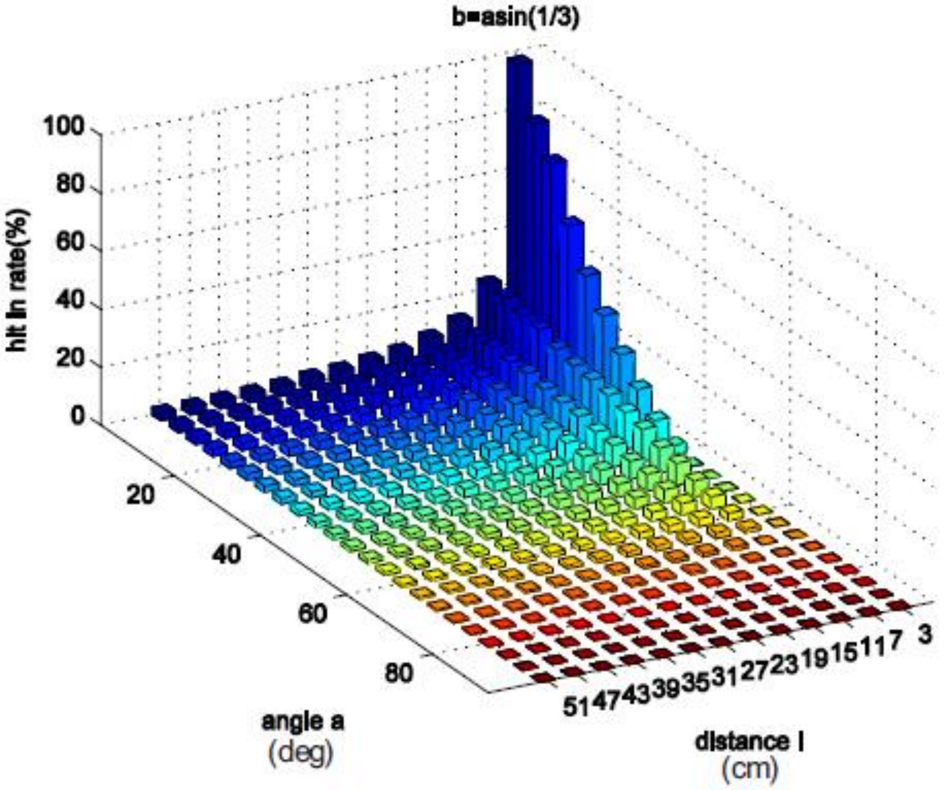


Figure 15 Hit-in rate as function of collision angle ***a*** and distance ***l*** for high-proficiency player with guidance.

From Table 2, the three players exhibit three different levels of proficiency, as shown from the maximum and average hit-in rates, while playing without guidance lines. The maximum hit-in rate ranges from 5.5% to 28%. The measure of similarity to the analysis ranges from 55.7 to 65.64 for low- to high-proficiency players, respectively. From Table 3, the maximum hit-in rate ranges from 82% to 97%. The measure of similarity to the calculated tolerance angle ranges from 82.69% to 95.2% for low- to high-proficiency players, respectively. This result indicates that all players benefit from our proposed visual guidance system in enhancing their skills. All exhibit enhanced hit-in rate, both in maximum values and average values, as shown in Table 4, but the low-skill player shows the maximum enhancement in skill with the help of our system. Furthermore, the enhancement of the three players' measure of similarity to the analysis tolerance angle profile also shows the reliability of our system in providing better assistance. The highly similar distribution pattern of the hit-in rate to that of the analyzed tolerance angle of the player with high proficiency shows the correctness of the analysis results. This evidence also shows that our vision tracking system and front end guiding interface are fully integrated and constitute a reliable and precise system as a whole.

Table 2 Performance comparisons of players at different proficiency levels without the authors' interactive guidance system.

Proficiency	Maximum hit-in rate (%)	Average hit-in rate (%)	D_A (%)	Similarity to analysis results (%)
1	5.5	3.18	44.28	55.71
2	9.5	3.41	40.52	59.48
3	28	3.84	34.35	65.64

Table 3 Performance comparisons of players at different proficiency levels using the authors' interactive guidance system.

Proficiency	Maximum hit-in rate (%)	Average hit-in rate (%)	D_A (%)	Similarity to analysis results (%)
1	82	4.88	17.30	82.69
2	85	5.1167	13.22	86.77
3	97	5.34	4.795	95.204

Table 4 Performance enhancement percentage between Tables 2 and 3 without and with the authors' interactive guiding system

Proficiency	Maximum hit-in rate enhancement (%)	Average hit-in rate enhancement (%)	Similarity to analysis results enhancement (%)
1	93	34.83	32.62
2	88.4	29.3	31.45
3	71.13	28.08	31.05

4.2 System Setup and Operation Processes

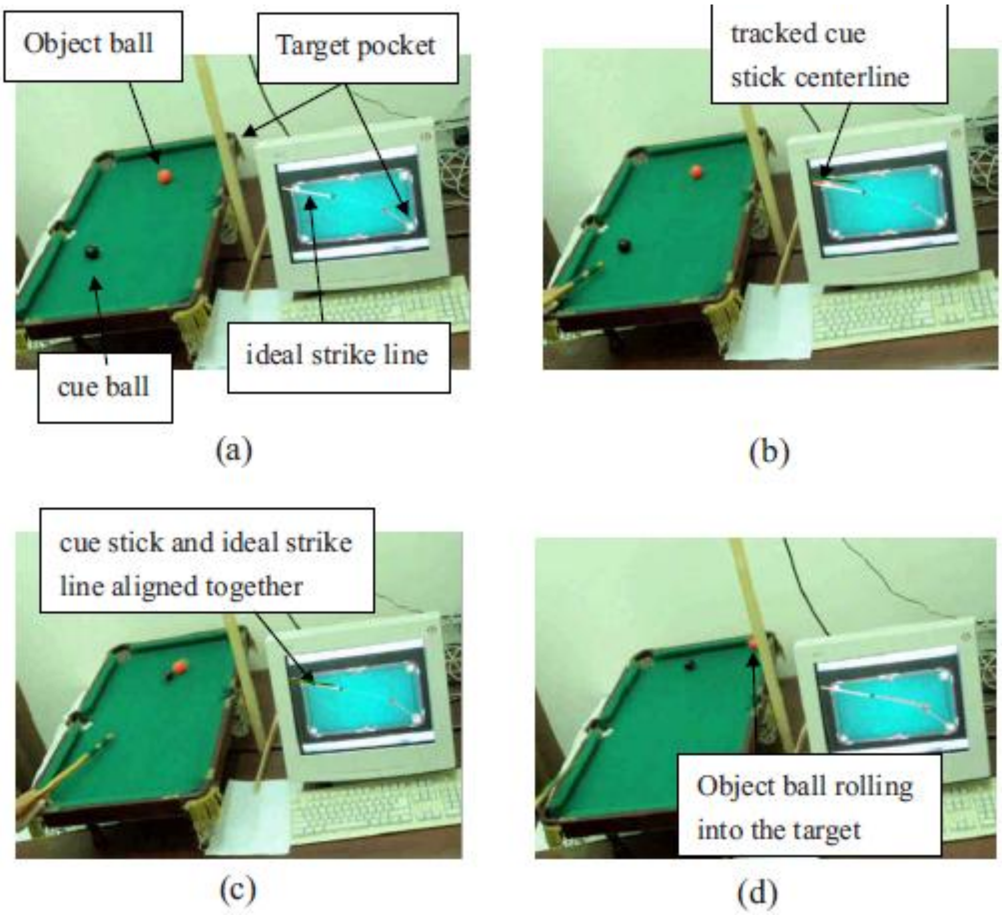


Figure 16 System setup and operation processes

Figure 15 shows the result of one stroke among the many exercise shots of our experiment for verifying the effects of tolerance angle on hit-in rate. The left-hand side of Fig. 15(a) shows the presence of one cue ball and one object ball on the pool table. On the right-hand side of Fig. 15(a), the center positions of both balls are precisely calculated by the image-processing algorithms. The balls' images are then superimposed at their respective positions on the empty table picture on the visual display of the PC. Given the relative pocket, object ball, and cue ball locations on the pool table, the strike guidance line can then be calculated according to the process in Sec. 3.2 and drawn from the cue ball as a light green line, shown on the PC on the right-hand side of Fig. 15(a). When the user starts to move his cue stick within the view range of the camera sensor above the pool table, as on the left-hand side of Fig. 15(b), the vision system starts to capture and analyze the cue stick image and display its location as a dark green line as shown on the right-hand side of Fig. 15(b). The user then starts to adjust the cue stick on the pool table, while watching the visual display. The goal is to align the dark green line with the light green line, which is in the ideal striking direction. Once the two lines match, the user can then start the stroke and drive the cue ball toward the object ball. Figure 15(c) shows that step. The user has aligned his cue stick with the instruction line on the right-hand side of Fig. 15(c). He then moves the cue stick toward the object ball. The cue ball can be seen to roll toward the object ball on the left-hand side of Fig. 15(c). After being hit by the cue ball, the object ball rolls into the target pocket as seen on the right-hand side of Fig. 15(d).

Chapter 5 Conclusion and Future Works

A novel vision-based billiard ball tracking system is combined with an interactive visual learning system to provide users with both a learning and an entertainment environment for the popular game. A least-squares error calibration method correctly correlates the actual locations of the cue and balls with the pixel coordinates in the visual display system. The major goal is to increase the aiming accuracy during the driving process to help improve the skills required to be proficient in the game and to increase the fun of playing it, without a complex electronic or mechanical setup on or around the playing table.

The vision system can not only trace but also recognize the balls and cue stick. The pixel information of the cue ball and object ball is used to display their images in a visual graphics interface. This pixel information is correlated with real pool table positions through a least-squares error transformation. The calculation of the theoretical guidance line is based on the actual positions of the cue, the object, and the target pocket. The calculated real-world positions and orientation of the guidance line are then transformed back to the corresponding pixel coordinates on the visual display and serve as a guide to the users for playing the game.

The selected object ball has been tested extensively, based on various geometrical parameters, with and without using our integrated system. Players with different proficiency levels were selected for the experiment. The results indicate that all players benefit from our proposed visual guidance system in enhancing their skills. All exhibit enhanced hit-in rate, both in maximum values and in average values, but the low-skill player shows the maximum enhancement in skill with the help of our system. The experiment results on hit-in rates show a consistent pattern with that of the analysis. The hit-in rate is thus tightly connected with the analyzed tolerance angles for sinking object balls into a target pocket. This proves the efficiency of our system.

This research has successfully integrated the vision system with an actual pool-playing environment. It not only provides an interactive learning environment to increase pool skills, but also increases the fun of this game with a minimum of hardware investment. Furthermore, the motion analysis results can aid in a smart game-playing strategy where the user can select the best shots among the many combinations of object ball and pocket locations.

Our next stage of research will include analysis of effects of fixing other parameters while varying the other two parameters. Since the tolerance angle has been found to be tightly related to the sink-in rate, the possible constraints these geometry parameters impose on the tolerance angle will be investigated. We are also investigating more sophisticated methods for strategy and planning, which would consider higher-order effects such as rebounds, multiple collisions, and preparatory placement for subsequent shots. Given careful analysis of the geometry parameters, it is possible to estimate the best locations to place a rebound cue and object balls for best next shots. The analysis will be conducted in a recursive manner. Some assumptions on friction and the magnitude of the striking force will be made. Data for this purpose can be derived from actual pool table experiments and used in both simulated and actual strike actions. Finally, the player's performance can be optimized and the sport can be fully enjoyed.

Reference

1. S. C. Chua, E. K. Wong, A. W. C. Tan, and V. C. Koo, “Decision algorithm for pool using fuzzy system,” in Proc. Int. Conf. on Artificial Intelligence in Engineering & Technology (ICAIET 2002), pp. 370–375 2002 .

2. M. Jouaneh and P. Carnevale, “The development of an autonomous robotic system for playing mini-golf,” IEEE Rob. Autom. Mag. 10 2 , 56–60 2003 .

3. K. Hashimoto and T. Noritsugu, “Modeling and control of robotic yoyo with visual feedback,” in Proc. IEEE Int. Conf. on Robotics and Automation, Vol. 3, pp. 2650–2655 1996 .

4. H. Nakai, Y. Taniguchi, M. Uenohara, T. Yoshimi, H. Ogawa, F. Ozaki, J. Oaki, H. Sato, Y. Asari, K. Maeda, H. Banba, T. Okada, K. Tatsuno, E. Tanaka, O. Yamaguchi, and M. Tachimori, “Volleyball playing robot,” in Proc. IEEE Int. Conf. on Robotics and Automation, Vol. 2, pp. 1083–1089 1998 .

5. J. Hoffman and E. Malstrom, “Teaching a miniature robotic manipulator to play chess,” Robotica 1 4 , 197–203 1983 .

6. F. C. A. Groen, G. A. den Boer, A. van Inge, and R. Stam, “Chess playing robot,” IEEE Trans. Instrum. Meas. 41 6 , 911–914 1992 .

7. L. Acosta, J. J. Rodrigo, J. A. Mendez, G. N. Marichal, and M. Sigut, “Ping-pong player prototype: a PC-based, low-cost, ping-pong robot,” IEEE Rob. Autom. Mag. 10 4 , 44–52 2003 .

8. S. C. Chua, E. K. Wong, and V. C. Koo, “Pool balls identification and calibration for a pool robot,” in Proc. Int. Conf. on Robotics, Vision, Information and Signal Processing (ROVISP 2003), pp. 312–315 2003 .

9. H. Nakama, I. Takaesu, and H. Tokashiki, “Basic study on development of shooting mechanism for billiard robot” in Japanese , Nippon Kikai Gakkai

Robotikusu, Mekatoronikusu Koenkai Koen Ronbunshu 2001 1 , 1A1.F8 1 -1A1.F8 2 2001 .

10. T. Jebara, C. Eyster, I. Weaver, T. Starner, and A. Pentland, “Stochastics: augmenting the billiard experience with probabilistic vision and wearable computers,” in Proc. Int. Symp. on Wearable Computers, pp. 138–145 1997 .

11. Z.-M. Lin, “The study of a billiard robot,” Thesis, Dep. of Mechanical and Electromechanical Engineering, Tamkan Univ., Taiwan 2003 .

12. B. R. Cheng, J. T. Li, and J. S. Yang, “Design of the neural-fuzzy compensator for a billiard robot,” in Proc. 2004 IEEE Int. Conf. on Networking, Sensing & Control, 2004 .

13. L. B. Larsen, P. M. Jensen, K. Kammersgaard, and L. Kromann, “The automated pool trainer—a multimodal system for learning the game of pool,” in Proc. Int. Conf. on Intelligent Multimedia and Distance Education 2001 .