A Vision Based Billiard Ball Tracking Platform, Motion Analysis Strategy And Their Integration For A Reliable Billiard Training System *

CHIHHSIONG SHIH, TANG-KUN LIU, YUANFAN YANG
Department of Computer Science and Information Engineering
Tunghai University
Taichung city, Taiwan R.O.C.
shihc@thu.edu.tw

A billiard ball tracking system is designed to combine with a visual guide interface to instruct users for a reliable strike. The integrated system runs on a PC platform. The system makes use of a vision system for cue ball, object ball locations and cue stick tracking. A least square error calibration process correlates the real world and the virtual world pool ball coordinates for a precise guidance line calculation. Users are able to adjust the cue stick on the pool table according to the visual guidance line instruction displayed on a PC monitor. The ideal visual guiding line extended from the cue ball is calculated based on a collision motion analysis. 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 slide into a pocket decide the difficulties of a strike. This value further depends on the distance from pocket to the object, the distance from object to the cue ball, and the angle between these two vectors. Simulation results of tolerance angles as function of these quantities are conducted. Selected object ball is tested extensively based on various geometry parameters using our integrated system. The experiment result as evaluated in hit-in rate has shown a consistent pattern with that of the analysis. The hit-in rate is thus tightly connected with the analyzed tolerance angles to sink object balls into a target pocket. This proves the reliability of our system, and the analysis results can be used for an efficient game playing strategy.

Keywords: real-time system, imprecise computation task, preemptive scheduling, mandatory subtask, optional subtask, total error, imprecisely scheduled tasks, NP-hard

1. INTRODUCTION

The game of billiards is a popular game, which has been played for hundreds of years in some form or fashion. 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 amount of practice. As this can be frustrating and unfruitful for beginners, and at times even for the 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 virtual environments.

Koo applied the concept of fuzzy logic to develop the decision algorithm for a best shot on computer generated pool environments [l]. Recently, researches have been carried out to create intelligent robotics in many applications [11-17]. This may include robot golf [11], yoyo [12], volleyball [13], chess [14, 15], and ping pong [16].

Some published papers show parts of the above functions of a billiard robot. Koo demonstrated the balls identification and calibration for a pool robot by the image processing technique [4].

Nakama developed a shooting mechanism for a billiard robot by a precise position mechanism [5]. A wearable computer and augmented reality has been demonstrated to help players to enhance the game of billiards by Jebara [6]. A vision algorithm is implemented which 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 each possible shot is ranked in order to determine its relative 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 human being [6, 7, 8]. Yang [8] has designed a billiard robot to imitate the learning ability of human beings to play billiard. The objective is to design a Neural-Fuzzy compensator for this billiard robot to improve the billiard skill. First, the predictable hitting error model is developed based on the recorded database of pocketing processes. Then, the predictable error is compensated by the Fuzzy controller to decide the cutting angle (hitting point) of the object ball automatically.

L.B. Larsen [9] describes the Automated Pool Trainer (APT), a multi modal pool training system developed at Aalborg University. The philosophy of the system is to automate the learning process. It is a multi modal system, utilizing 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 and automatic motion selection strategy supporting the system. A human expert is behind the system to plan the training courses. Also the laser pointer can poise visual harm to personnel under training.

As such, these devices are often not constructive to the learning process. Many are not popular in the 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 guide methods.

A novel vision based billiard ball tracking system is designed to provide player an interactive guiding system for a user 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 complex electronic or mechanical setup on or around the playing table. The system achieves the goal by tracking the actual balls 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 objective ball and target pocket locations.

The target pocket and object ball are selected on the visual display system based on a motion analysis algorithm for a best shot. This imaginary guiding line is calculated based on the basic physics law 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 on the display system. User then adjusts this center line by moving the cue stick around to match that of the calculated ideal collision line from the cue ball. Once both lines align together with each other on the visual display, 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 a motion analysis. The motion analysis has shown that the tolerance angle around a nominal aiming direction decide the success rate to sink 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.

Table 1. comparisons of relevant pool learning tools and setups

Index	**Vision tracking**		**Motion Analysis Strategy**		**Service target**		**Visual guide instruction media**		
Work	**Ball center**	**Cue stick**	**Fuzzy Logic**	**Tol. Error Model**	**robot**	**human**	**PC Monitor**	**LCD Eye goggle**	**Laser light pointer**
Koo[1]	no	no	yes	no	no	no	no	no	no
Koo4	yes	no	no	no	yes	no	no	no	no
Jebara[6]	yes	no	no	yes	no	yes	no	yes	no
Yang[8]	yes	no	yes	no	yes	no	no	no	no
Larsen[9]	yes	no	no	no	no	yes	no	no	yes
Nakama[5]	yes	no	no	no	yes	no	no	no	no
Shih	yes	yes	no	yes	no	yes	yes	no	no

This research first presents the system setup and the visual interface, followed by the image calibration process to accurately correlate the ball center pixel coordinates to the actual positions on the billiard table. Then, the algorithms developed in processing and recognizing the images during the tracking process of the cue stick, the cue and object balls is discussed. Analysis is followed 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. Theoretical model is built to understand the interrelation between them. Finally, an extensive testing drive of the whole system including back end tracking and front end visual display is exercised to verify the accuracy of the system by changing these parameters. We change different combination 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 experiment results are then compared with the analysis models. Important issues and future works are then presented.

2. SYSTEM DESCRIPTION

The experimental results show that the improvement both in memory utilization and in decoding time was proportional to *d*. Hence, for pre-processed data, such as images compressed using the JPEG method, which have very small *d* (1 or 2), the performance improvement will be minor. However, for ordinary data, the experimental results reveal that the memory utilization for CMRPC can be improved from 1.1% (=(2168-2144) bits/2168 bits) to 4.1% (=(2168-2068) bits/2168 bits); the decoding time can be improved from approximately 16.2% (=(1.91-1.60) sec/1.91 sec) to 25.5% (=(2.12-1.58) sec/2.12 sec). It is also observed that the proposed modified method is superior to the one-shift method and the table-lookup method [3].

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 board table by a set of fixtures. The camera orientation is set arbitrarily directly above the pool table. The only requirement is that the field of view must cover the whole billiard board 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, leaving two isolated regions in 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 which instructs 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.

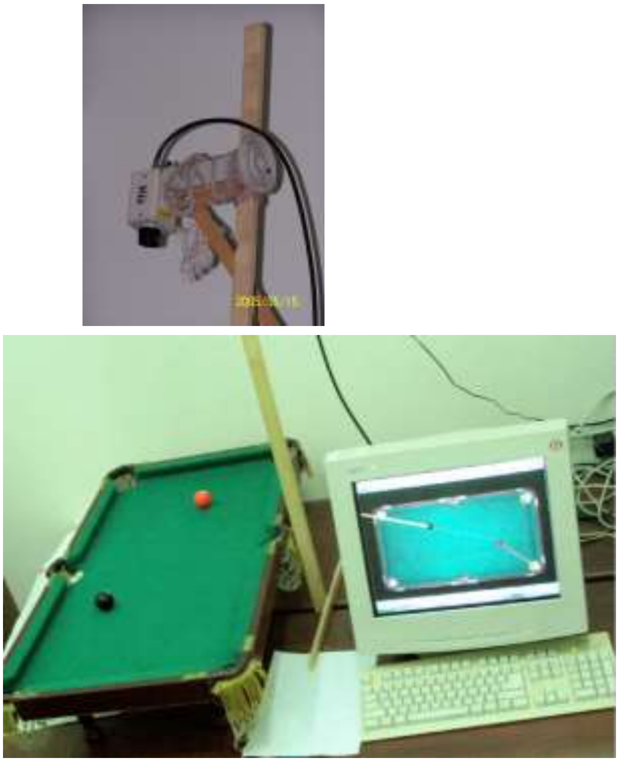
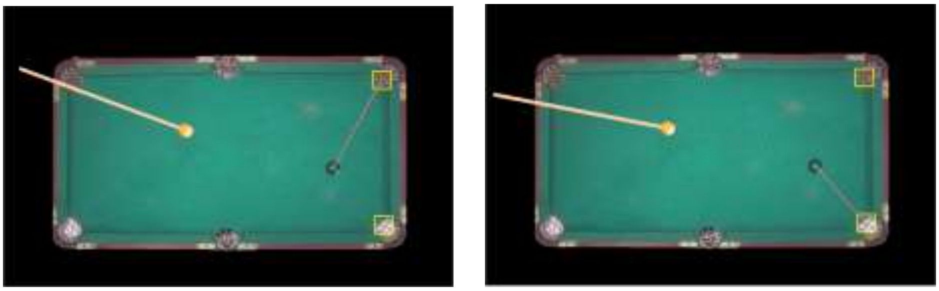


Fig. 1 System setup and configuration

3. LEARNING MODEL AND VISUAL DISPLAY

The visual display system plays the role of instructing a striking strategy of the cue stick to the users. Given combination of different target pocket and object ball selected by the motion analysis results in section 5, a correct aiming direction has been 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 a basic physics law 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 one point alone the extension line which is exactly one ball diameter, ‘d’, away from the object ball center as marked in the Fig. 3. Next 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.

The selection of different pocket and object ball combination 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 toward different direction given the same cue ball and object ball locations with different target pockets locations.



(a)

(b)

Fig. 2. Visual interface showing calculated guidance line

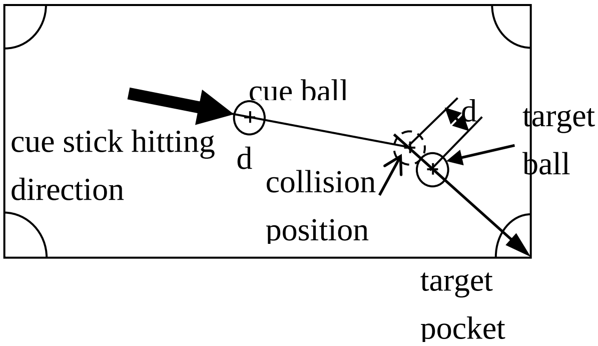


Fig. 3. Guiding line model

4. 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 grid pattern is printed and placed on the billiard table as shown in Fig. 4(a). The dimension of this mini-pool table is in the order of 40x80 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 153x2 matrix, A. $A = \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ \bullet & \bullet \\ x_{153} & y_{153} \end{bmatrix}$, where (xn,yn), n=1~153, is the nth grid point coordinates on the calibration board of the billiard table. The unit is cm. The image coordinates of the grid intersection points are extracted using a regular thinning and edge extraction image processing algorithms as in Fig. 4(c). The image coordinates are then stored in another 153x3 matrix, B. $B = \begin{bmatrix} ix_1 & iy_1 & 1 \\ ix_2 & iy_2 & 1 \\ \bullet & \bullet & \bullet \\ ix_{153} & iy_{153} & 1 \end{bmatrix}$,where (ixn,iyn), n=1~153, is the nth grid image pixel coordinates. A correlation matrix, T, is then calculated using a least square error transformation equation as shown below.

T=inverse(B’*B)(B’*A). Given this transformation matrix(3x2), the image coordinates are transformed back to the real world coordinates on the pool table via the following equation: A’= B*T; and superimposed together with the original coordinate matrix, ’A’ to form Fig. 4(d). The standard deviation error between original and transformed coordinates for each grid point is in the order of 0.1 cm.

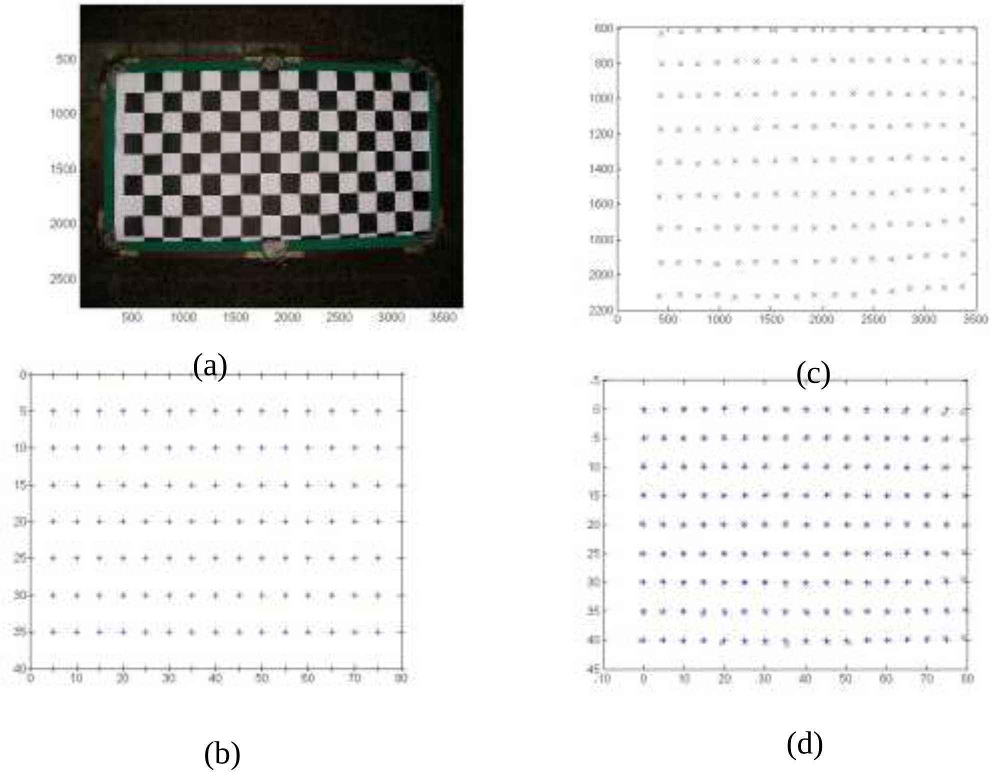
The image processing steps need to first calculate the centroid position of the cue ball and the target ball. Fig. 5(a) is a raw image of the billiard table including a cue ball and an object ball. An RGB to HSV conversion is exercised for each pixel as in equation (1). A histogram chart is then built based on the H values for all the pixels

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

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

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

of an empty table image. A range of H values from 110 to 185 is selected to collect the table pixels, since most of table cloth pixels exhibit these values and posses the highest peaks in a histogram diagram of H value versus number of pixels from the image of Fig. 5(b). The unwanted pixels including those of the environment background and possibly the balls and stick images can then be filtered out. An initial result is shown in Fig. 5(b). A median filter algorithm is then followed 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 (e). The boundary pixels of each independent object are further extracted by a recursion traversing process that groups them into different sets. A region scanning process is then followed inside each of the separated sets to identify the color of each of the independent island. This is to decide if it belongs to a color ball or a stick tip. Tracking on the cue stick is accomplished by tracing the two separated colored regions on the tip of the stick.



(a)

(c)

(b)

(d)

Fig. 4. Calibration charts and superimposed error map

The back end vision system and front end visual display has been fully integrated with this prototype mini pool table as shown in Fig. 8. The data flow between the different sub-systems is illustrated in Fig. 6. 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 visual display system and the coordinate transformation step ‘a’ as shown in Fig. 6. Step ‘a’ converts the image coordinates into world coordinates on pool table through a transformation matrix in the calibration process. Step ‘b’ calculate the theoretical strike line using the world coordinates of the cues, and pocket based on the collision model and analysis results. The derived center line is then converted back to the visionary 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 stoke to sink the object ball into selected pockets.

5. MOTION ANALYSIS STRATEGY

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 don't involve any collisions with other objects. If no solid balls are found, we test the ball for possible trajectories.

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 shall assume that the force with which the cue ball is hit is going to be determined by the user). So, what is fundamentally being suggested to the user is an angle at which to hit a cue ball, say angle *c* as a deviation from the line connecting the cue ball to the solid object ball. The more accuracy we need on angle *c*, the harder the shot. Fig. 7 shows how we can determine the required angle, using equation (4),(5),and (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 angle *a* at the solid ball. We can compute angle *c* using this information as we11 as a bound on the maximum error on angle *c* which is angle *d*.

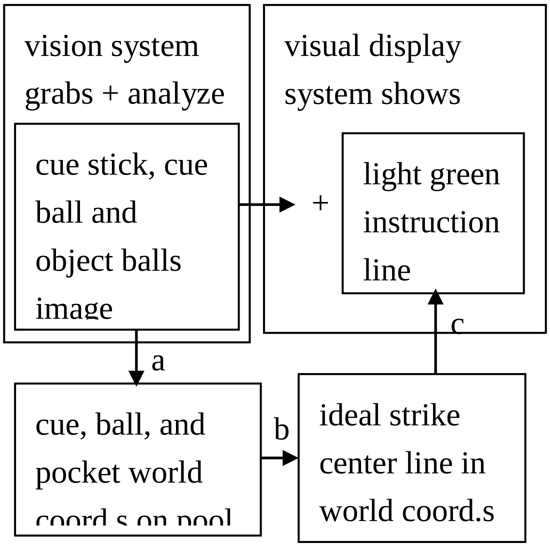
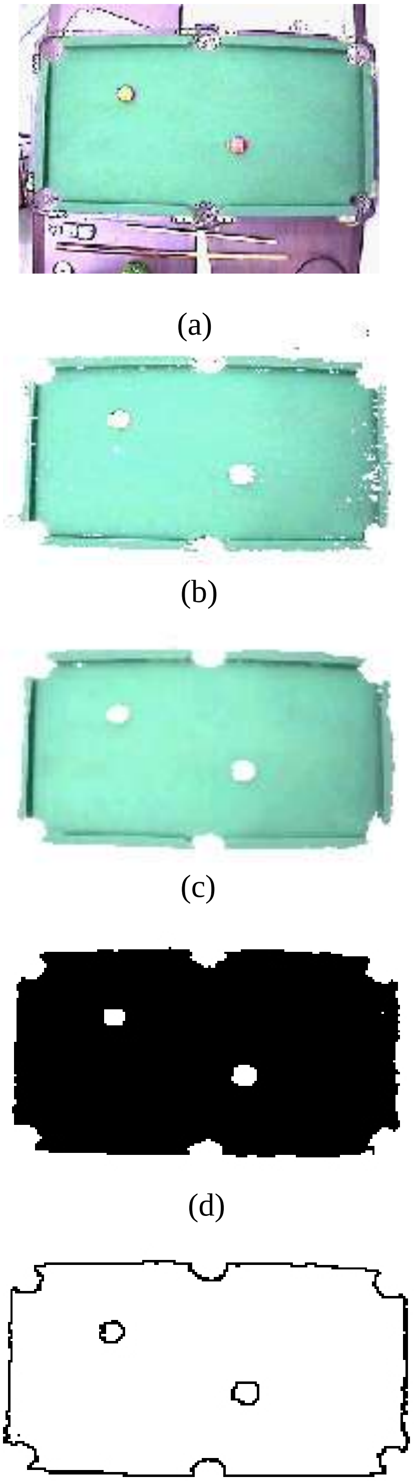

vision system grabs + analyze
cue stick, cue ball and object balls image
visual display system shows
+
light green instruction line
a
c
cue, ball, and pocket world coord.s on pool
b
ideal strike center line in world coord.s

Fig. 6. Integration data flow

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


(a)

(b)

(c)

(d)

(e)

Fig. 5. Boundary edge extractions

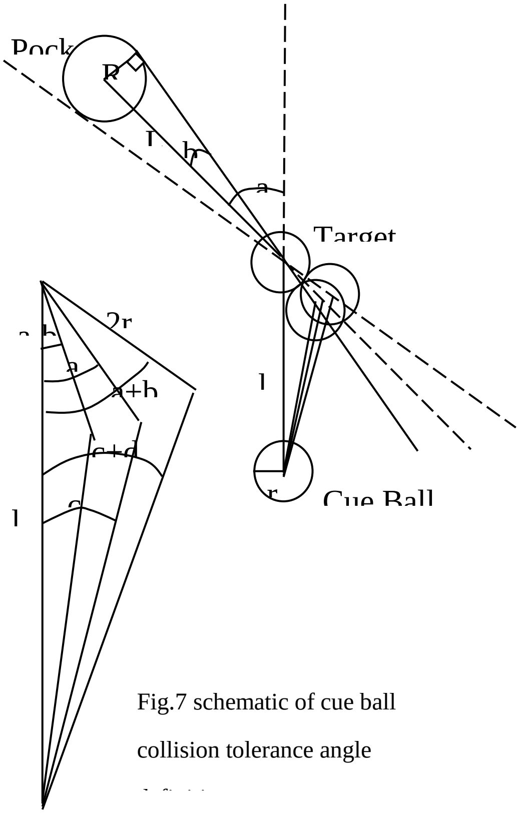


Fig.7 schematic of cue ball collision tolerance angle

$$b = a\sin(R/L) \qquad (1)$$

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

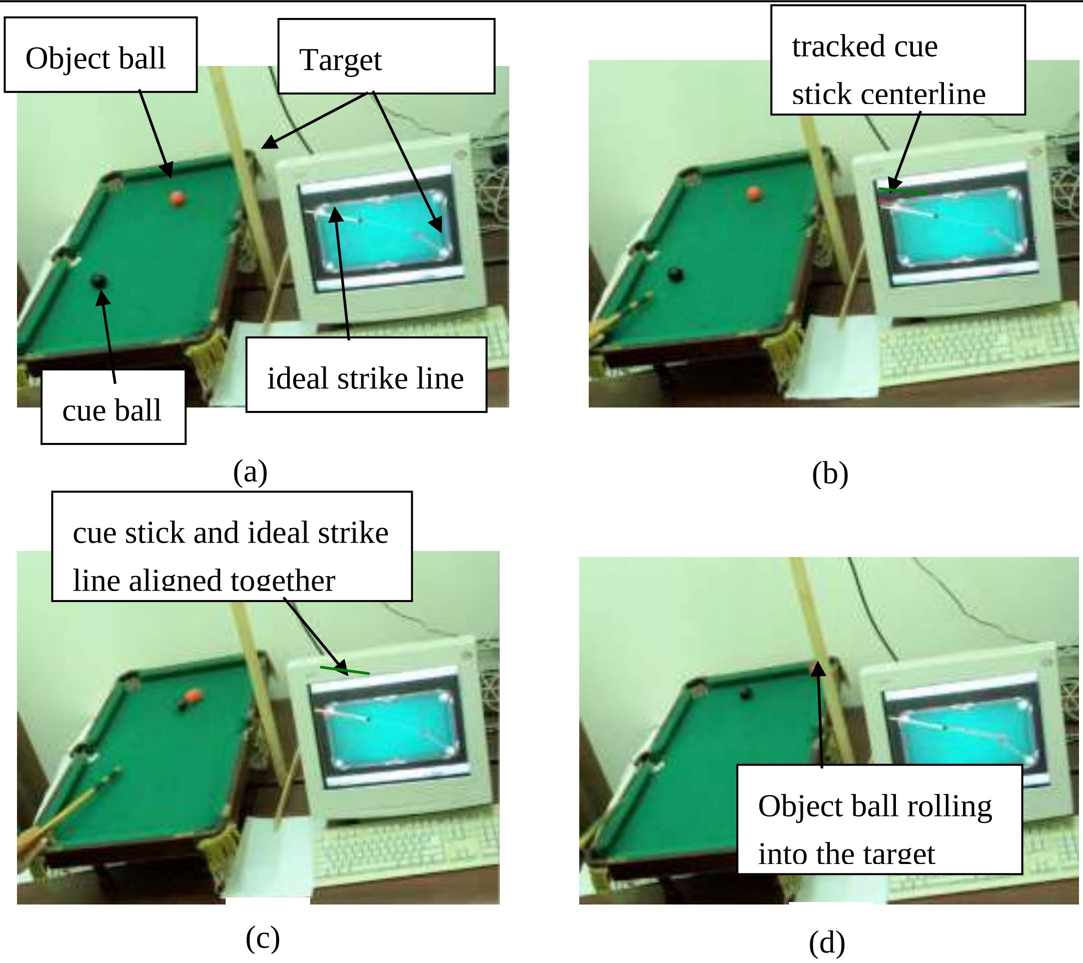
Based on this analysis, the collision tolerance angle d is in fact a function of l, L, and a. The larger the tolerance angle is, the easier for the player to sink the object ball. Given a fixed L value(a 1/3 ratio value), this paper first evaluate the tolerance angle value on the different combination of l and a values each ranging from 0 to 50 and from 0 to 90 respectively. The other paper will discuss the effects of fixing other parameters while varying the other two parameters on the tolerance angles. The l value is in fact a multiple of radius of the billiard ball (r). Fig. 8 is the results of such calculations. The profiles are intuitively reasonable with the maximum tolerance at zero ‘a’ angle and about one ball diameter distance between the cue and object balls. Generally, this value decreases gradually with increasing l values at a constant ‘a’ value. The minor reduced tolerance angle values at close range with the object balls for larger ‘a’ angles has been found to be due to the nature of equation 6.

6. EXPERIMENT RESULTS

We then test the hit-in rate at each combination of cue ball angle and distance variation between cue and object balls. The object ball is fixed at a distance about three times that of the pocket radius. At each combination, we calculate the ideal visual guidance line using the tracked center points from the vision system. Fifteen shots were played based on the guidance line. The success sink in rate is then calculated counting the total number of shots. The results are again plotted again as a function of l and a as in Fig. 9. There are variations between different combinations. However, the profile distribution is approximately the same as the analysis simulation results. This evidence shows our vision tracking system and calibration procedure are a reliable and precise system.

$$d = a\sin\left(\frac{2r\sin(a+b)}{\sqrt{4r^2+l^2-4rl\cos(a+b)}}\right)-c \qquad (3)$$

Fig. 10 is the result of one stroke of the many exercise shots of our experiment for verifying the effects


Object ball
Target
cue ball
ideal strike line
tracked cue stick centerline
cue stick and ideal strike line aligned together
Object ball rolling into the target

(a)

(b)

(c)

(d)

Fig. 10 System setup and operation processes

of tolerance angle on hit-in rate. Left hand side of Fig. 10(a) shows the existence of one cue ball and one object ball on the pool table. On the right hand side of the Fig. 10(a), the center positions of both balls are precisely calculated by the image processing algorithms. The balls photo images are then superimposed at the respective positions on the empty table picture of 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 and drawn from the cue ball as a light green line shown in the PC in the right hand side of Fig. 10(a). When user starts to move his cue stick within the view range of the camera sensor above the pool table as in the left hand side of Fig. 10(b), 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. 10(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 the ideal striking direction. Once both lines match with each other, user can then start the stroke and drive the cue ball toward the object ball. Fig.10(c) shows such step. The user has aligned his cue stick with the instruction line on the right hand side of Fig. 10(c). He then moves the cue stick toward the object ball. The cue ball can be seen to roll toward object ball on the left hand side of Fig. 10(c). After colliding with the cue ball, the object ball is seen to roll into the target pocket as seen in the right hand side of Fig. 10(d).

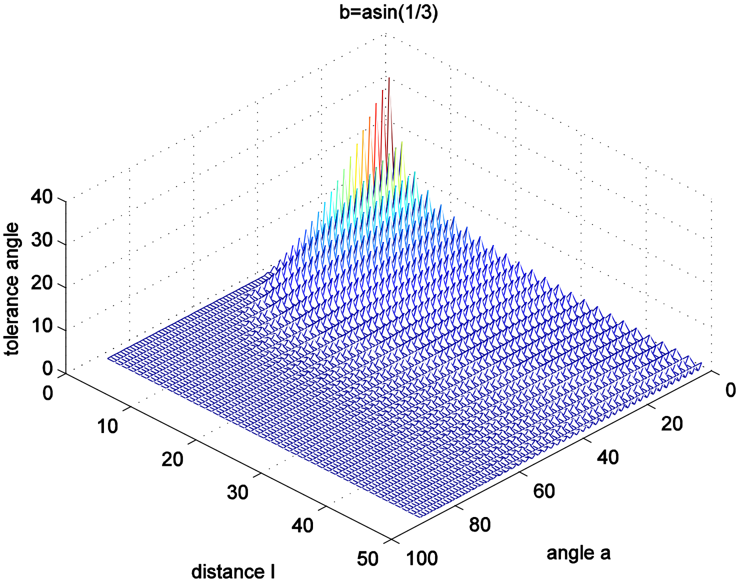


Fig. 8 Calculated cue ball tolerance angle d profile

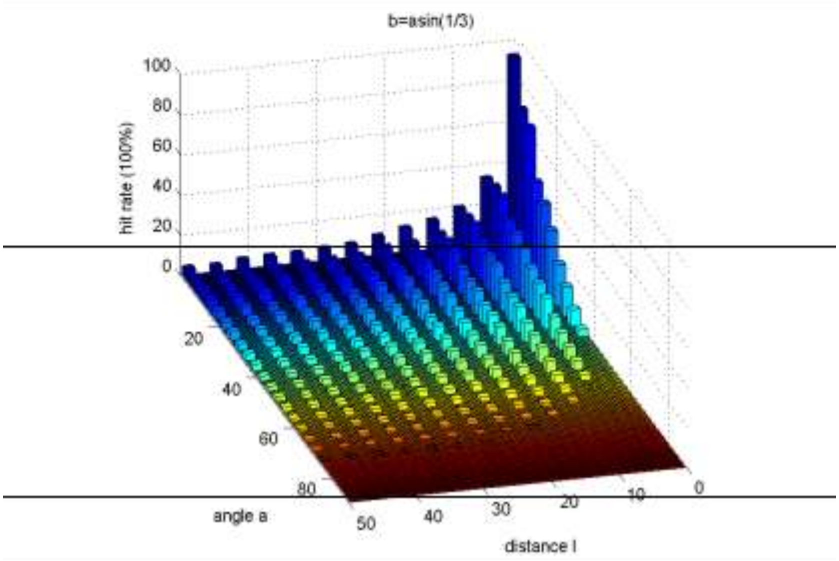


Fig. 9 Hit-in rate as function of collision angle *a* and distance *l*

7. 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 entertainment environment for the popular billiard ball game. A least square error calibration method correctly correlate the actual cue and balls geometry locations with the pixel coordinates in the visual display system. The major goal is to increase the aiming accuracy during the hitting process to help expedite the skills required to be proficient in the game and to increase the fun of playing this game without complex electronic or mechanical setup on or around the playing table.

The vision system can not only trace but also recognize ball objects and cue stick. The pixel information of the cue ball and object ball are used to display their corresponding photo images in a visual graphics interface. These pixels information are correlated to a real pool table positions through a least square error transformation process. The calculation of the theoretical guidance line is based on the actual positions of cue, object and target pockets. 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.

Experiment results have shown successes in verifying the aiming tolerance error model based on motion analysis. This research has successfully integrated the vision system with an actual pool playing environment. It not only provides an interactive learning environment to increase the pool skills but also increases the fun of this game with a minimum of hardware investment. Further more, the motion analysis results can aid in a smart game playing strategy where user can select the best shots among the many objects balls and pockets locations combinations.

Our next stage of research would include analysis of effects of fixing other parameters while varying the other two parameters on the tolerance angles. Since the tolerance angle has been found to be tightly related to the sink-in rate, the possible constraints of these geometry parameters have 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 as well as preparatory placement for subsequent shots. Given careful analysis on the geometry parameters, it is possible to estimate the best locations to place a rebound cue and object balls for next best shots. The analysis will be conducted in a recursion manner. Some presumption on frictions and exertion of striking force would be made. These data can be derived from actual pool table experiment, and used in both simulation and actual strike actions. Finally, the pool game playing performance can be optimized and the fun of the sport can be fully enjoyed.

REFERENCES

1. S. C. Chua, E. K. Wong, Alan W. C. Tan, and V C. Koo, “Decision Algorithm for Pool using Fuzzy System, ” *Proceedings of the lnternational Conference on Artificial Intelligence in Engineering & Technology* ICAIET 2002, Kola Kinabalu, Malaysia, pp. 370-375, 17-18 June 2002.
2. T. Braunl “Research Relevance of Mobile Robot Competitions”. *IEEE Robotics & Automation Magazine*: pp.32-37, 1999.
3. K. Kiguchi, K. Watanabe, and T. Fukuda, ”Design of a Human-Like Desired Grasping Force Planner for Object Manipulation by Robot Manipulators,” *Machine Intelligence Robotic Control,*Vo1.3,No.4.pp.167-173, 2002.
4. S. C. Chua, E. K. Wong and V. C. Koo, “Pool Balls Identification and Calibration for a Pool Robot”, *Proceedings of the lnternational Conference on Robotics, Vision, Information and Signal Processing* (ROVISP 2003), Penang, Malaysia, pp. 312-315, January 2003.
5. H. Nakama, 1. Takaesu and H. Tokashiki, “Basic Study on Development of Shooting Mechanism for Billiard Robot”, *JSME, Robotic Workshop,* IAIF8, 2001.
6. T. Jebara, C. Eyster, 1. Weaver. T. Starner and A. Pentland, “Stochastics: Augmenting the Billiard Experience with Probabilistic Vision and Wearable Computers”. *In Proceedings of the International Symposium on Wearable Computers, Cambridge*, Massachusetts, pp. 138-145, 1997.
7. Zhi-Ming Lin, “The Study of A Billiard Robot”, *Thesis*, Department of Mechanical

and Electro Mechanical Engineering, TKU, Taiwan, R.O.C., 2003.

8. B.R. Cheng, J. T. Li, J. S. Yang, “Design of the Neural-Fuzzy Compensator for a Billiard Robot”, *Proceedings of the 2004 IEEE International Conference on Networking, Sensing & Control*, Taipei, Taiwan, March 21-23, 2004.
9. P. Grogono, Mathematics for Snooker Simulation, *personal communication*, 2001.
10. L.B. Larsen, P.M. Jensen, K. Kammersgaard, L. Kromann, “The automated pool trainer—a multi modal system for learning the game of pool,” *Proceeding of the International Conference on Intelligent Multimedia and Distance Education*, 2001.
11. M. Jouaneh, P. Carnevale, “The development of an autonomous robotic system for playing mini-golf,” *IEEE Robot. Autom*. Mag. 10 (2) (2003) 56–60.
12. K. Hashimoto, T. Noritsugu, “Modeling and control of robotic yoyo with visual feedback,“ *Proceedings of the IEEE International Conference on Robotics and Automation vol. 3* (1996) 2650–2655.
13. 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. Vamaguchi, M. Tacyhimon, “Volleyball playing robot”, *Proceedings of the IEEE International Conference on Robotics and Automation vol. 2* (1998) 1083–1089.
14. J. Hoffman, E. Malstrom, “Teaching a miniature robotic manipulator to play chess,” *Robotica* 1 (4) (1983) 197–203.
15. F.C.A. Groen, G.A. den Boer, A. van Inge, R. Stam, “Chess playing robot,” *IEEE Trans. Instrumen. Measure*. 41 (6) (1992) 911–914.
16. L. Acosta, J.J. Rodrigo, J.A. Mendez, G.N. Marichal, M. Sigut, “Ping-pong player prototype: a pc-based, low-cost, ping-pong robot,” *IEEE Robot. Autom.* Mag. 10 (4) (2003) 44–52.

Chun-Chao Yeh (葉春超) received his Ph.D. and M.S. degrees in computer science and information engineering from National Taiwan University, Taipei, Taiwan, in 1998 and 1991, respectively, and his B.S. degree in computer science from National Chiao Tung University, Hsinchu, Taiwan, in 1989. While he was working toward his Ph.D. degree at National Taiwan University, he got a DAAD/NSC scholarship to be a visiting researcher in GMD, Germany, from July 1996 to April 1997. He has been an Assistant Processor of Computer Science Department, National Taiwan Ocean University, Keelung, Taiwan, since 2000. His research interests include computer networks, real-time and embedded systems, and web servers.

Jie-Yong Juang (莊志洋) received the B.S. degree in Electrical Engineering from National Taiwan University, Taipei, Taiwan, in 1976, the M.S. degree in Computer Science from University of Nebraska, Lincoln, in 1981, and the Ph.D. degree in Electrical Engineering from Purdue University, West Lafayette, IN, in 1985. He is a Professor of the Department of Computer Science and Information Engineering of the Taichung Healthcare and Management University, Taichung, Taiwan. From February 1991 to July 2000, he was a Professor of the Department of Computer Science and Information Engineering of National Taiwan University, Taipei, Taiwan. He was with the Department of Electrical Engineering and Computer Science of Northwestern University, Evanston, IL, from January 1985 to January 1991. His areas of research include multimedia and communication, embedded OS.