J Intell Robot Syst (2012) 67:25–41
DOI 10.1007/s10846-011-9639-4

REVIEW PAPER

Billiard Combat Modeling and Simulation Based on Optimal Cue Placement Control and Strategic Planning

Chihhsiong Shih · Chorng-Shiuh Koong · Pao-Ann Hsiung

Received: 14 July 2010 / Accepted: 25 October 2011 / Published online: 9 December 2011

Abstract The selection of a best sequential shots for a given start cue position is a major challenging task in a billiard game. A new algorithm is proposed as a strategy to apply maximum tolerance angle search sequentially. The strategy considers combinations among all pockets and target object balls during both the pre and post collision shots selection processes. A simulation program is developed to test the strategy in a competition scenario by players with different proficiencies. The level of proficiency of players in the competition is controlled by a threshold value as a criterion to evaluate capability to conduct consecutive shots and when to give out right of play. The winning score of each game (win rate) is used as a performance comparison index among different gaming situations and to verify the effectiveness of the algorithm. The initial results of several simulation games using our strategy show that higher proficiency player can out beat lower proficiency player easily. This is consistent with the gaming situation in the real world, showing the consistency of our simulation program. The simulation also verifies that the play order does decide the final competition outcomes, when the players' proficiencies are close to each other. This work is the first to investigate the effects of consecutive shots and order of play on the billiard gaming results. A low cost training system is proposed to verify the efficiency of the repositioning algorithm in real world settings. The system adapts an augmented reality technology to instruct users for reliable aiming assistance. It makes use of a vision system for cue ball, object ball locations and cue stick velocity tracking. In all, the simulation program can provide an initial proof of the effectiveness of the reposition algorithm in the competition situation. Experiments results of maximum tolerance angle all pocket search strategy using our training facility as tested by users with different skill levels all out performed the results without guidance for the set of users with the same proficiency.

C. Shih (✉)
Department of Computer Science,
Tunghai University, No. 181 Section 3,
Taichung Harbor Road, Taichung, Taiwan,
Republic of China
e-mail: shihc@go.thu.edu.tw

C.-S. Koong
Dept. of Computer and Information Science,
National Taichung University, Taichung, Taiwan,
Republic of China

P.-A. Hsiung
Dept. of Computer Science and Information
Engineering, National Chung Cheng University,
Chiayi, Taiwan, Republic of China

Keywords Human-computer interface · Intelligent tutoring systems · Interactive learning environments · Teaching/learning strategies

1 Introduction

In a real world pool game such as 8-ball pool, 9-ball pool, and snooker, the rule to win a game varied a lot. Given the variety of winning requirements, however, there is one common feature between them. That is the higher the number of sequential sinks the player is able to render, the higher chance the player will win the game. The selection of a best sequential shots for a given start cue position is not an easy task in a billiard game. Additional to the start shot on the current existing object balls on a table, the repositioning of the cue after its collision with an object ball decides its success on successive shots.

A new strategy is proposed to apply maximum tolerance angle search sequentially. The strategy considers combinations among all pockets and target object balls during both the pre and post collision shots selection processes. Using this strategy, an estimated travel distance is evaluated for best possible successive shots. Given such distance on the table, an initial velocity to drive the cue stick is inversely calculated. A simulation program is developed to test the strategy per users with different proficiency in the real gaming situation. The level of proficiency of players in the competition is controlled by a threshold value as a criterion to evaluate the right of play in the sequence of shots selection process. The win rate of each game is used as a performance comparison index among different gaming situations and to verify the effectiveness of the algorithm. The initial results of several simulation games using our strategy show that higher proficiency player can out beat lower proficiency player easily. This is consistent with the gaming situation in the real world. In this case, the order of play doesn't influence the gaming results at all.

In the case of players with relatively close proficiency, higher skill player actually loses the game to the low skill player. The order of player does have impact on the competition results. For the cases where the players have equal proficiency the play order decides who win the game. Generally, the one who plays first wins the game. This work is the first to investigate the effects of order of play on the billiard gaming results.

In the experiment, a billiard training system is devised to integrate with a vision guidance system for both aiming and stroke control on a target object ball guided by our cue reposition strategy. The ideal strike velocity will be shown on the visual display as a guide for users to drive the actual cue stick consistently. Our tracking system makes use of an augmented reality technique for stroke instructions, when hitting the cue ball. The win rate of several real games among players with and without using our training system is used as an index for comparing the performances. Experiment results of all pockets search strategy using our training facility as tested by users with different skill levels all out performed the results without guidance for the set of users with the same proficiency. This not only proves the reliability of our training device, but also proves the effectiveness of the all pocket search algorithm in guiding users for optimal performance.

2 Relevant Works

Many billiard video games and robots are devoted to be an opposing counter part to the player. This includes robot golf [4], yoyo [5] volleyball [6], chess [7, 8], and ping pong [9] just to name a few. In this category, researches have focused on creating intelligent robotics for entertainment purposes. Various simulations and analysis have been exercised on different popular billiard games such as 8-ball, 9-ball and snookers [2, 10–12]. Smith [10] applied artificial intelligence on the 8-ball game playing strategy development. A program, PickPocket, was developed based on a traditional search framework, familiar to games such as chess, adapted to the continuous stochastic domain of billiards. Experimental results presented exploring properties of two search algorithms, Monte-Carlo search and Probabilistic search. Greenspan [11, 12] further analyzed the physics nature of billiard motions and collisions and developed a library as a physics engine for actual game playing. They all contributed to the correct simulation and prediction of game playing results. However, none of them actually contribute to the real world billiards gaming environment. The smart strategy and precise analysis have no method to help the user enhance the enjoyment of the game.

The other category of the billiard system aimed to assist players in real billiard game training. Jebara [1] demonstrated that a wearable computer and augmented reality helps players enhance the game of billiards. This system was similar to ours in that players received the strike instructions in real time and can be applied to enforce a precise stroke. However, the delay in the head mounted display can cause dizziness for users during motion. The cost is higher due to the use of LCD goggles as a head mounted display. Also the repositioning strategy is not discussed and used to help users. Larsen [3] described the Automated Pool Trainer (APT), a multi model pool training system developed at Aalborg University. It is a multi model system, utilizing spoken interaction combined with a graphical output and a computer controlled laser pointer for user communication. There is no error analysis and automatic strike selection strategy supporting this system. A human expert is behind the system to plan the training courses. The use of human expertise not only limits the robustness of the training but also increases the cost of the system compared to what this study is proposing.

This research first presents the system setup and the visual interface in Section 3. The tolerance angle gives a measure of how hard it is to sink an object ball as stated in [13] and restated again in Section 4. Basic rule of post collision motion of a cue with an object ball will influence the final position for next optimal shot and is discussed in Section 5. The repositioning algorithms are discussed to find out the set of best post collision cue positions via Section 6. The competition simulation algorithm is also added to test the proposed repositioning algorithm. Initial simulation results are also illustrated in this section. Calibration of the ball deceleration as function of the friction coefficients and relevant factors are explored in Section 7.1. An extensive suite of the competition simulation based on various players' proficiency is conducted in Section 7.2. Statistics are provided to prove the effectiveness of the repositioning strategy. Finally, a testing drive of the whole system including back end tracking and front end visual display is exercised to verify the accuracy of the system. The interrelation of theoretical analysis model from Sections 5 and 6 with real world repositioning play profile by users at different skill levels is illustrated in Section 7.3. Important issues and summaries are then presented in conclusion.

3 System Description

The guidance system is modified from a vision tracking system [13] and a PC running a visual display of the captured image of pool table in real time with the addition of cue stick velocity estimation and aiming direction 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 cue stick doesn't need to be tagged as that in previous design [14]. A Pentium III PC running a graphical user interface sits right next to the billiard table. The graphical interface displays the pool table images captured in real time.

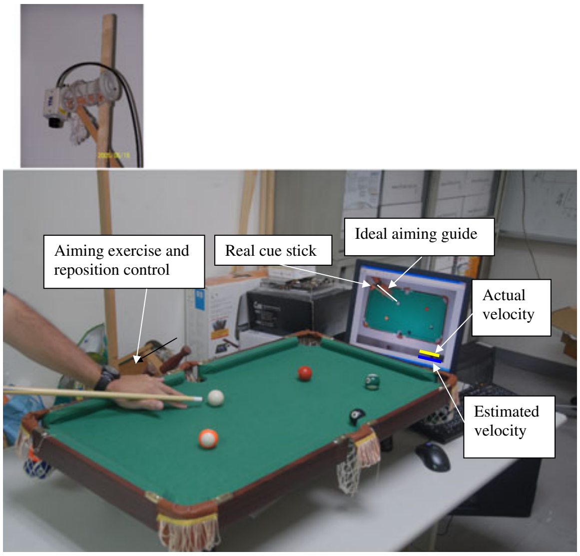


Fig. 1 System setup

The software in the PC executes both the visual display which instructs user how to place the cue stick on the pool table through an augmented reality technology and the processing of pool table images captured by the CCD sensor in real time. Given the analysis results from back end simulation about the orientation of the cue stick and the required velocity to reposition the cue to its next best position, user then moves the real cue stick to align with the guidance line and strike with proper force to sink the selected target object balls into the target pocket and reposition the cue to a desired position best for next shots.

4 Shot Repositioning Difficulty Measure

The criterion we use to optimize the sequential shots in the combat scenario is based on a difficulty measure in the aiming process. This criterion has been discussed in [13] and is briefly stated here.

$$b = a\sin(R/L) \tag{1}$$

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

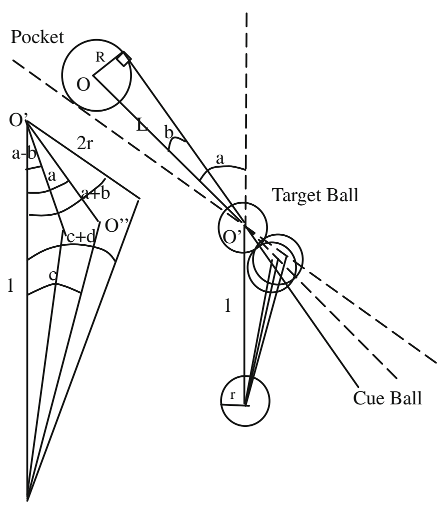


Fig. 2 Schematic of cue ball collision tolerance angle definition

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

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. Figure 2 shows how we can determine the required angle, using Eqs. 1, 2 and 3. 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 L and l is angle 'a' at the intersection at the center. We can compute angle c using this information as well as a bound on the maximum error on angle c which is angle d. To precisely pocket an object ball to the center, the cue ball must aim at the position of O'' according to the ghost ball theory. Given a known angle a value, side 2r and l distance, Eq. 2 calculates the angle c required to send the cue ball from point P to O'' using the well known two-side-one-angle rule to calculate the opposite angle of angle a. With the angle a added for by angle b, the angle of $c + d$ can be calculated by Eq. 3 using the same rule as above. Finally, the tolerance angle, d, can be derived by subtracting angle c.

For each possible shot, we compute angle d and find an optimal shot with the maximum d value. This is a simplified first-order strategy model since it is only 2D and does not include spin effects, kinematics or rebound.

5 Minimum Post Collision Distance Estimation

The goal of our selection strategy is to sink a best selected object ball into a proper pocket and get the post collision cue ready for subsequent shots. We make use of idealized linear collision physics to predict the possible rest position of the cue ball after its collision with a selected object ball. Here we care about the selection of a minimum initial speed of the cue stick to drive cue to hit a selected object ball and to sink it into a selected pocket based on a search process in Section 6. Given such an initial velocity, the object ball will travel just enough to sink into the target pocket and the cue will be deflected and stop at a fixed location. For a cue to stop at an optimal position best for next strike, the search has to go beyond this position. This initial speed depends on the travel distance from the object position to the pocket and from cue to the object. Given this minimum speed, it is possible to estimate the post collision position of a cue. To find an optimal speed of the cue stick to drive the object ball into a target pocket and to send the cue to an optimal position for next best shot, additional speed (or force) is needed as shown in Fig. 3. We describe the procedure to find the minimum post collision position of a cue ball with an initial speed on object ball just enough to roll toward a target pocket and stop at the pocket center with zero speed.

The basics of collision physics and relevant notations are shown in Fig. 3 as quoted from [14]. From this figure, the known quantities are S and S_1. S is the distance from the start cue position to the ghost ball position, while S_1 is the distance from the object ball to the target pocket. The target quantities we are interested in are S_{min} and CV_{opt}. We use an inverse derivation process first to find the minimum speed required by an object ball to travel to the target pocket and stop at center of the pocket, P_{min}. This value can be solved with Eq. 4. Given this minimum speed of object ball, P_{min}, the minimum initial speed of the cue after colliding with the object ball, V_{min}, can be derived from Eq. 5 using the 90 degree rule of basic physics as shown in Fig. 3 where angle "*a*" being the cut angle. Then the post collision travel distance of the cue ball, S_{min}, can be solved using Eq. 6, assuming a constant deceleration of the ball motion from the constant friction coefficient between the ball and the table cloth. The ending velocity of cue right before collision, CV_{end}, corresponds to the hypotenuse while the starting velocity after collision, V_{min}, corresponds to the leg opposing the cutting angle. One would speculate the minimum post collision travel distance would vary as a function of the cue driving force. After substituting in V_{min} of Eq. 4 in 6 to derive Eq. 7, it actually shows that this quantity is function of S_1 and 'a' only as shown in Fig. 3 and Eq. 7. To calculate the minimum speed to drive the cue to force the collided object to sink into the target pocket, we use the relation of starting velocity, CV_{min}, with the ending velocity, CV_{end}, as given in Eq. 8. After rearranging terms and substituting in CV_{end} in Eq. 9, the required minimum initial cue speed is expressed as function of S_{min} and S and cut angle '*a*' as in Eq. 10. The speed that can place the cue to the optimal post collision position can then be expressed by Eq. 11 as function of S_{opt} and S.

$$P_{\min} = \sqrt{2 * \mu * S_1} \tag{4}$$

$$V_{\min} = P_{\min} * \tan(a) \tag{5}$$

$$S_{\min} = \frac{V_{\min}^2}{2 * \mu} \tag{6}$$

$$S_{\min} = \frac{(P_{\min} * \tan(a))^2}{2 * \mu} = \frac{\left(\sqrt{2 * \mu * S_1} * \tan(a)\right)^2}{2 * \mu} = S_1 * \tan(a)^2 \tag{7}$$

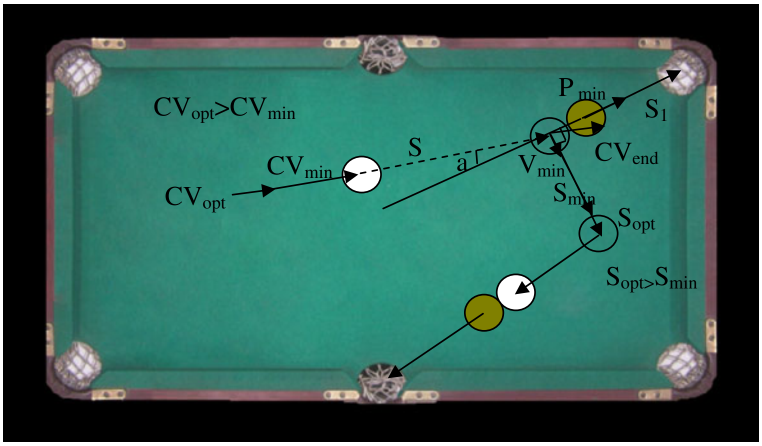


Fig. 3 Post collision physics schematic and notations

$$CV_{\min}^2 = CV_{end}^2 + 2 * \mu * S \tag{8}$$

$$CV_{end} = \frac{V_{\min}}{\sin(a)} \tag{9}$$

$$CV_{\min}^2 = \frac{2 * \mu * S_{\min}}{(\sin(a))^2} + 2 * \mu * S \tag{10}$$

$$CV_{\text{opt}}^2 = \frac{2 * \mu * S_{\text{opt}}}{(\sin(a))^2} + 2 * \mu * S \tag{11}$$

6 Optimal Cue Repositioning Control Algorithm and Competition Simulation

The repositioning of cue ball has decisive effects on the follow up shots. One extreme example is to place the cue close to object balls very close to the jaw of nearby pockets. This makes it very easy to sink the object ball, while the other extreme case is when the cue is far away from an object ball. We propose a search algorithm to locate an optimal position of cue for best follow up shots. Figure 4 gives the details of the optimal search algorithm. Following the analysis results from previous section, the cue will pick a best object ball and pocket combination for its first shot based on a maximum error tolerance criterion. This corresponds to step 1 in Fig. 4. Given this selected object and pocket combination, the direction of motion and the minimum travel distance of cue post collision are decided from analysis results in the previous section. We then search alone the path starting from the minimum stop point of post collision motion, given that there still are objects on the table. This corresponds to the step 2 in Fig. 4. The search goal is to find a point on the path with maximum error tolerance angle. A threshold value is added to check if the search should continue if a maximum tolerance angle exceeds such threshold. This is to control the number of successive shots during a simulation scenario. The search is based on a uniform grid comparison of each evaluation results of the tolerance angle. On each grid point, the combination of all object balls with all accessible pockets is searched for a maximum tolerance angle. The resolution is around one hundredth of an inch. Once this value is larger than a stored maximum value, the new maximum replace the original maximum, and the object ball and the target pocket for next shot is stored. The

```
1. Given a start cue position, sCue, find a best first shot among all object balls based
   on max. tolerance angle aiming all accessible pockets
2. while there still are accessible object balls on the table and maxAng>threshold(set
   by the simulation program for different players in a competition)
3. set maxPK=nearest pocket ID,
4. set maxAng=0,
5. set curAkID =current attack object
   5.0 for each accessible pocket(clear path from object to pocket)
       5.1 find a post collision deflection path of cue to drive curAkID object
           toward the target pocket per 90 degree rule of section 5.
       5.2 given a minimum post collision cue starting position calculated by
           eq. (7) of section 5.
           5.2.1  given a cue position, for each accessible object ball and
                  pocket combination, calculate tolerance angle =ang
           5.2.2  if (maxAng < ang) then
                    maxAng =ang;
                    store objectID as MaxID;
                    store maxPK=current pocket ID;
                    store optCuePos = current cue position;
                  endif
           5.2.3  update cue position for a fixed amount of increment in the
                  direction of deflection, until new cue position reach table
                  edges
           5.2.4  repeat 5.2.0~5.2.3
   end for (of pocket traversal)
6. Given start cue position, scue, optCuePos(post collision optimal stop position),
curAkID and maxPK, inversely calculate the driving initial speed for optCuePos
using eq. 11 given S_opt and S, this is the estimated initial speed to be displayed
7. user drives the cue using the estimated speed and sinks a previous best candidate
object ball
8. mark the curAkID as invisible after actually sinking curAkID into pocket
9. mark the object ball with MaxID as current attack target, curAkID.
10. place the cue at optCuePos, repeat 2~9
11. end while loop, user can't continue shooting due to higher difficulty in the last
shot
12. randomly exert force on the last cue position and place the cue to another position
```

Fig. 4 Optimal repositioning algorithm considering all pockets

search ends at the path intersection with the table rail (edge). These correspond to the step 3, 4, 5 in Fig. 4. Once the optimal position is found, Eq. 11 is used to calculate the needed initial velocity to drive cue and stop at this position. This is the estimated velocity that will be drawn on the GUI for users to follow and place the cue to this position at his best. This is the step 6 of Fig. 4. After sinking the optimal candidate object ball into a selected pocket, maxPK, the cue will stop at another optimal position ready for next best shot. These steps correspond to step 7∼10 in Fig. 4. After the while loop ends due to failure to meet the threshold constraint for another optimal shot, the user has to shoot randomly and give up the right of play to the opponent. This is done in step 12 of Fig. 4. To compare the performance of algorithm of Fig. 4 used by players with different proficiency in a real gaming scenario, another algorithm is developed to exercise the algorithm in Fig. 4 given different threshold values simulating players with different proficiency. Figure 5 describes such algorithm in detail. The algorithm repeated exercises algorithm in Fig. 4 until table is cleared. Once the execution works in the loop of algorithm of Fig. 4, the shots will continue until user encounters a difficulty shot beyond his capability to complete the shot. This is quantified by the maximum tolerance angle being smaller than a selected threshold value. In such situation, the user usually misses his shot on selected target. Our algorithm simulates such situation by shooting randomly still at originally selected target object ball.

The all pocket algorithm is analyzed to have O(MNK) efficiency where M is the number of object balls and N is the number of post collision search points, and K is the number of pockets. The order of computation complexity is in the order

1. set threshold values for different players

2. While table is not cleared

3. execute algorithm in Fig. 4 for first player, using a specific threshold

4. after exit from the step 11 of Fig. 4, play random shot and turn the right of play to the next player; record the accumulated successful shots

5. execute algorithm in Fig. 4 for second player, using another specific threshold

6. after exit from the step 11 of Fig. 4, play random shot and turn the play of right to the previous player, record the accumulated successful shots

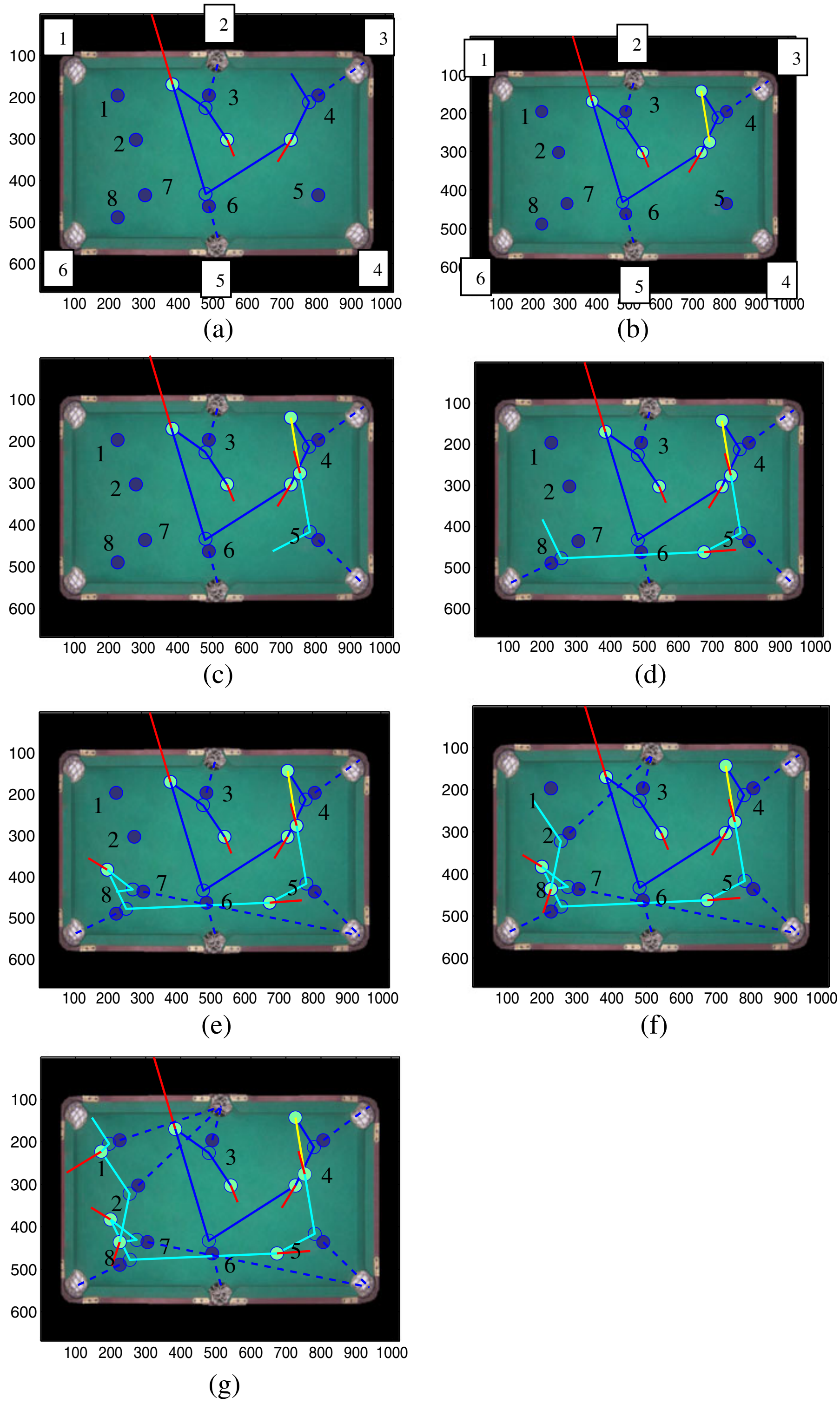
7. end while

Fig. 5 Simulated competition among different players applying optimized repositioning control with different proficiency (controlled by different threshold values)

of third degree polynomial. This is caused by the triple search operation of the nested loop on the post collision path for an optimal position best for follow up shots.

Figure 6 shows one sample run of algorithm of Fig. 5 with a randomly picked start cue position for a simulated combat situation between two players. There are eight object balls distributed around the table. Two players with different proficiency as presented by different threshold values are selected for testing the algorithm. A low proficiency player is presented using the threshold of 1, while another high proficiency player makes use of a threshold of 0.0001. This selection is intuitively reasonable for each player. As high threshold value selected means easy to miss a large tolerance angle shot and thus can represent a low skill player. The number of consecutive shots can be expected low. While a low threshold value means many tough shots with relatively low maximum tolerance angle can be rendered, the number of consecutive shot can be expected higher than player with higher threshold. The continuous path of successful shots sequence of the first player is marked blue, while that of the second player is marked white in Fig. 6. At the end of a successful sequence shots, the player usually misses shot and is represented by different colors. The missing shots of the first player are presented with yellow lines, while those of the second player are presented with black lines. After a player misses shot, the right of play is turned to another player. As marked in Fig. 6, the first shot picked by the first player is number 3 object ball which is closest to the cue and at the jaw of pocket 2. This gives an intuitively reasonable proof of the effectiveness of our algorithm in identifying the best first shot with the maximum error tolerance criterion. For the second optimal shot, the program picked ball number 6 instead of ball number 1, 2, 7 and 8. This also illustrated that we can filter the cue attack angle from the second stop of the cue with the object ball number 6 which is almost zero compared to those still on the table as in Fig. 6(a). Again proving the effectiveness of our approach, as experienced player knows that it is easier to pocket an object ball with a smaller attack angle. The next optimally selected object ball is number 4 ball sinking into pocket number 3. While after

Fig. 6 Simulated shots sequence of two players applying algorithm in Fig. 5



successfully sinking ball number 4, the cue stop position is close by the top rail, and makes the first player very hard to render any more successful shots. This is caused by the lower maximum tolerance angle than the threshold value with all other balls on the table. The player then makes a random shot at the possible target ball. The path is marked with yellow lines in Fig. 6(b). After missing a target ball, the right of play is then passed to the opponent, the second player. Since the second player is represented with a much lower threshold, many high difficulty ball and pocket configurations with small tolerance angles can be rendered. Thus the second player can successfully sink ball number 5, 8, 7, 2, 1 consecutively into pocket number 4, 6, 4, 2, 2. The sequence of shots can be viewed from Fig. 6(c)–(g). The table is then cleared, and player two claims the victory of the game. What is worth noting is that after sinking ball number 8 into pocket 6, the cue stops at a position that is almost linear between ball number 7 and pocket 4 in Fig. 6(e). This allows the second player to render the shot easily and also prepares the post collision cue position for the following shots on balls 2 and 1 all into pocket 2. This is an advantage of the algorithm in Fig. 4 by considering all pockets of each shot during the optimal search process for a maximum tolerance angle. This is also a positive influence that a player will regularly adopt in exercising a shot. Our algorithm has successfully integrated this feature within the search process.

7 Experiment Results

7.1 Ball Deceleration Calibration

The thrust on the cue stick of our guidance system has a decisive effect on the cue motion distance. We adapt the same calibration process as in [14]. About twenty shots of the dry run by a sophisticated user are recorded and shown in Fig. 7 to calibrate a corresponding characteristic curve of velocity to the travel distance pairs. We record the tracked velocity and corresponding travel distance of the hit ball for each shot. This way both the friction force and hand motion effect can be compounded in a single quantity. Users doing the

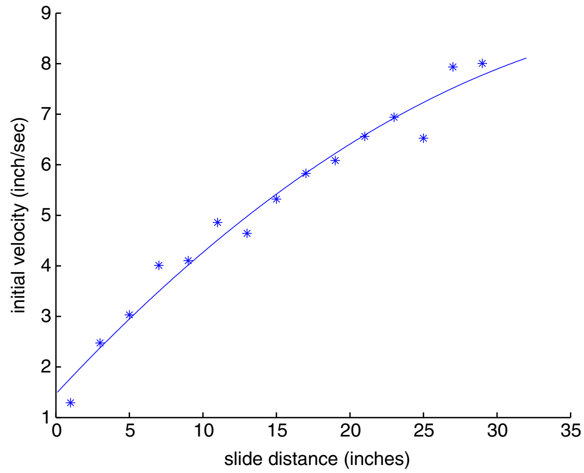


Fig. 7 Velocity measurements and calibration results

experiments need to move the stick as close to the guidance line as possible.

The derived deceleration rate will be substituted in Eq. 11 to get an estimation of the start speed to drive the cue to the optimal post collision position for next best strike. After a few dry runs of the cue stick stroke to the scale of estimated velocity, he or she can actually drive the cue, sink the best selected object ball into the target pocket and see the cue slide toward the analyzed optimal position where the tolerance angle is largest (Fig. 8).

7.2 Competition Simulation Results with Varying Thresholds

Since we are able to simulate the missing shots situations in algorithms in Figs. 4 and 5, the

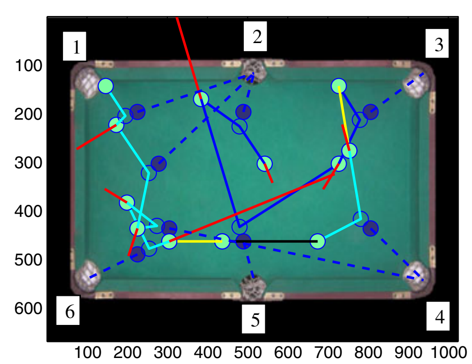


Fig. 8 Simulated sequence shots of two players with different thresholds (1 and 0.5) with 1 sinking three balls while 0.5 sinking five balls

competition among players with different proficiency is exercised. The proficiency is controlled by the threshold values of algorithms in Fig. 4. Again, higher threshold values represent users with lower proficiency, while lower threshold values represent users with higher proficiency. The

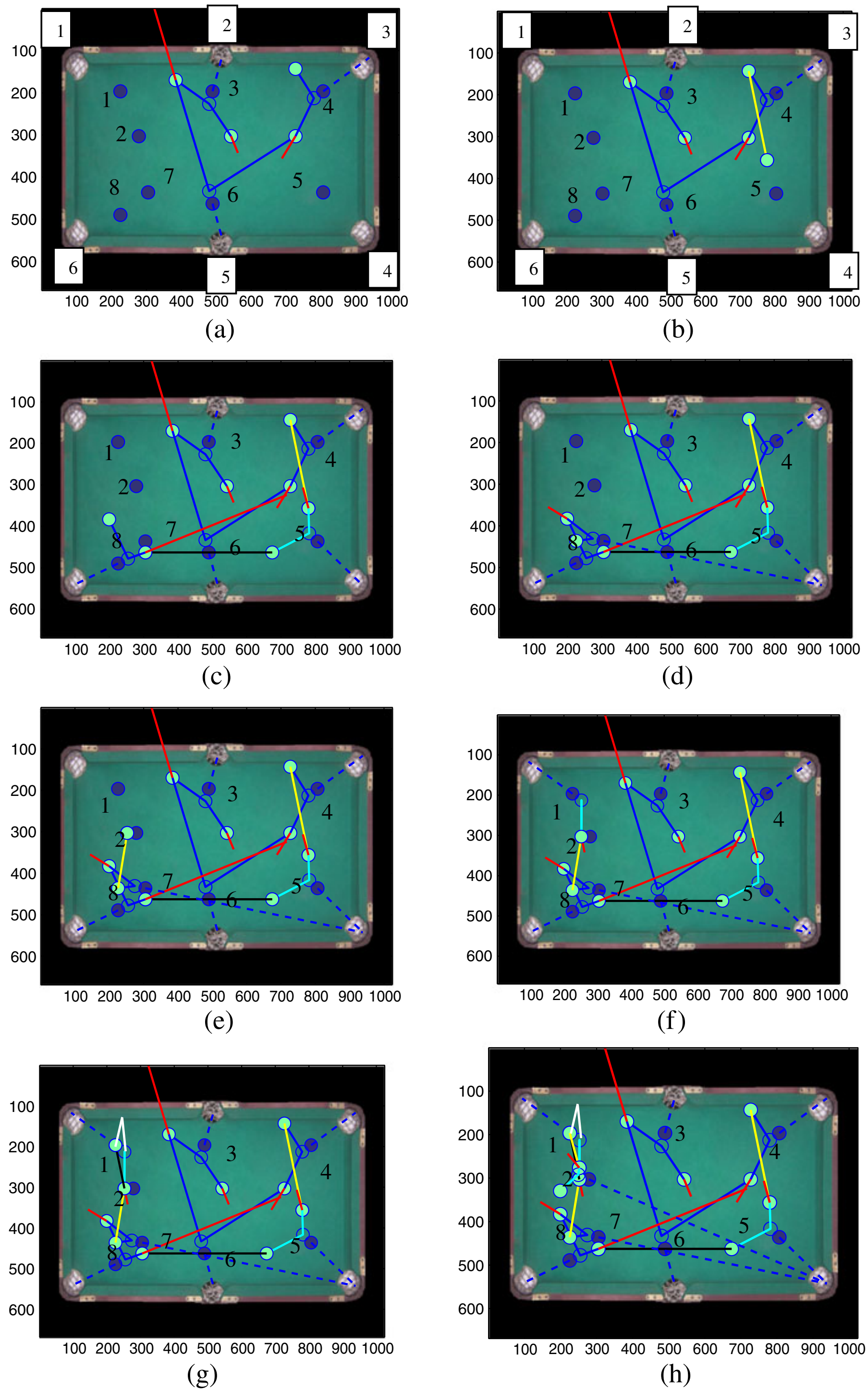


Fig. 9 Simulated sequence shots of two players with different thresholds (1 and 0.5) with 1 sinking five balls while 0.5 sinking three balls

threshold pairs we pick to experiment with the competition situation include 1 to 0.0001, 1 to 0.5, and 0.5 to 1. Ten games are simulated and the play statistics are collected. The primary statistics for performance comparison are the total number of successful shots and the win rate. The win rate is defined as the ratio of the total number of successful shots between the two competitors.

Figure 9 is the simulation results of two players in competition with threshold values of 1 and 0.5. The one with threshold value 1 plays first. This player misses his shot at the fourth strike as shown in Fig. 9(b). The second player follows and gains one point after sinking ball number 5 as marked by white line in Fig. 9(c). Then he misses the next shot as marked by the black line in Fig. 9(c), while aiming for ball number 8. The first player then takes over and scores two more points by sinking ball number 7 and 8 as shown in Fig. 9(d). Then he misses the next shot while aiming for ball number 2 as marked by the yellow line as shown in Fig. 9(e). The second player immediately continues the shot and scores ball number 1 and 2 as shown in Fig. 9(f). One interesting phenomenon observed from this competition game is that the first player with higher threshold actually wins the game, which seems in contradiction with the original assumption. The ideal situation is that the low threshold player should score more points than the high threshold player. Actually, this phenomenon is reasonable in that if the proficiency difference among the two players is small (in this case 0.5), the one who plays first usually can win the game. This is caused by the advantage of the first player gains in the first few successful shots. After scoring the first few shots by the first player, the two players pretty much compete on the same chance of missing shots and taking turns in gaining approximately equal points. Finally, the first player easily out runs the second player with the few balls he gains in the beginning. This phenomenon will be further investigated in the following experiments.

To further investigate the effect of the game playing order on the final results of a race, the same set of players as in the Fig. 9 is recalled for another race with switched order. The first player plays with 0.5 threshold value, while

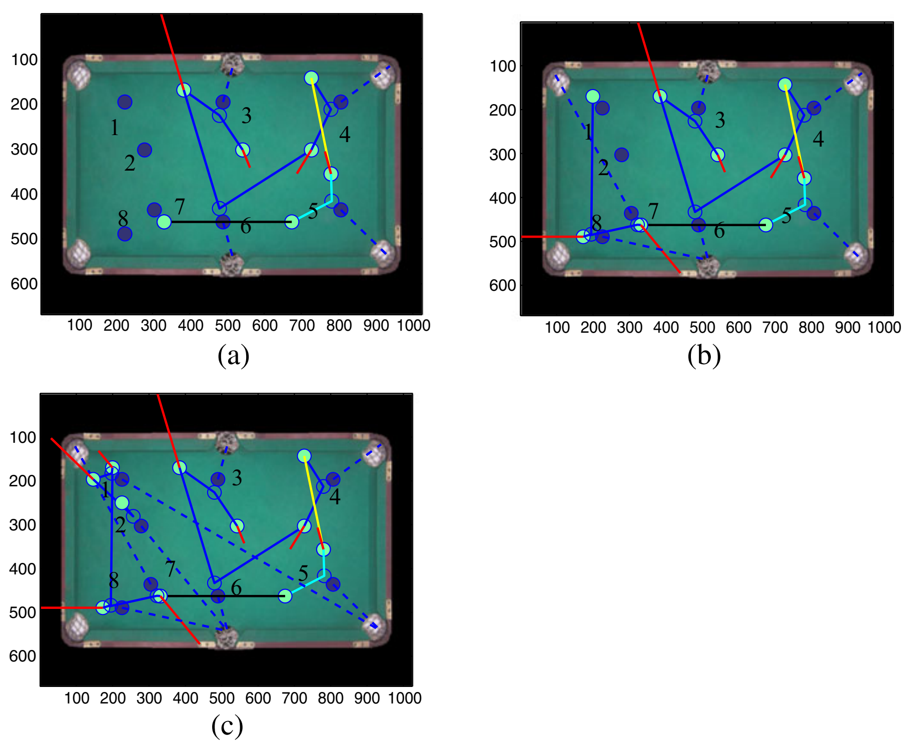


Fig. 10 Simulated sequence shots of two players with different thresholds (0.5 and 1) with 0.5 player playing first

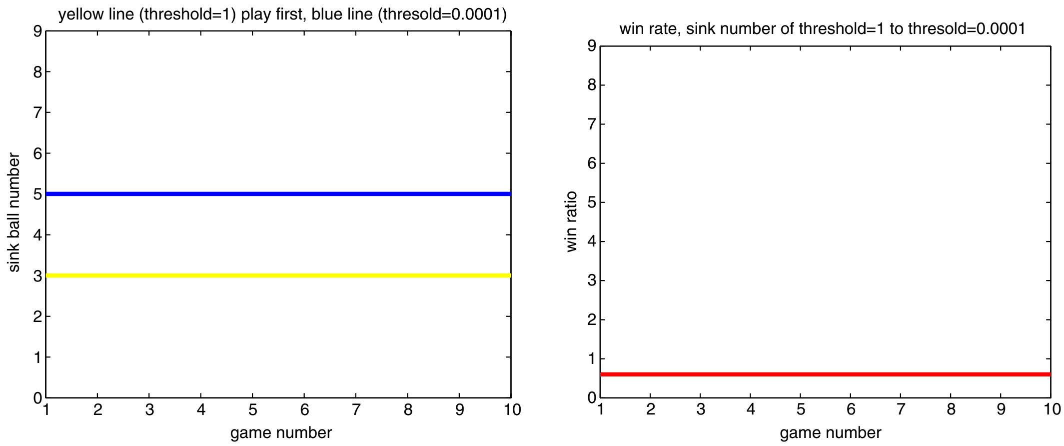


Fig. 11 Simulated ten games statistics of two players with high contrast threshold values (1 and 0.0001)

the second player plays with 1 threshold value. Figure 10 shows the results of one of such game. The first player scores the first three balls with the same sequence as in Fig. 9(a), then he misses the following shot aiming at ball number 5. This is a tough shot with large attack angle and far distance between cue and object balls. Both players fail to complete the stroke and have to give up right of play to the opponent. After sinking the ball number 5 by the second player, the player misses the shot aiming at ball number 8 as marked by black line in Fig. 10(a) and turns the right of play to the first player again. Due to the lower threshold value of the first player, he actually is able to sink ball 7, 8, 1 and 2 consecutively as in Fig. 10(b) and (c). Finally, the table is cleared and the first player wins the game by scoring a total of seven balls, while the second player with higher threshold can only sink one ball. This is an interesting phenomenon that shows the impact of order of play in wining a contest, given a relative small threshold value difference. In this case, the first player has only a slightly lower threshold value than the second player. In addition to this advantage of lower threshold value, the order of first shot helps make the big gains in the competition of the first player. The large win ratio was attributed to the first few successful shots gained by the first player over

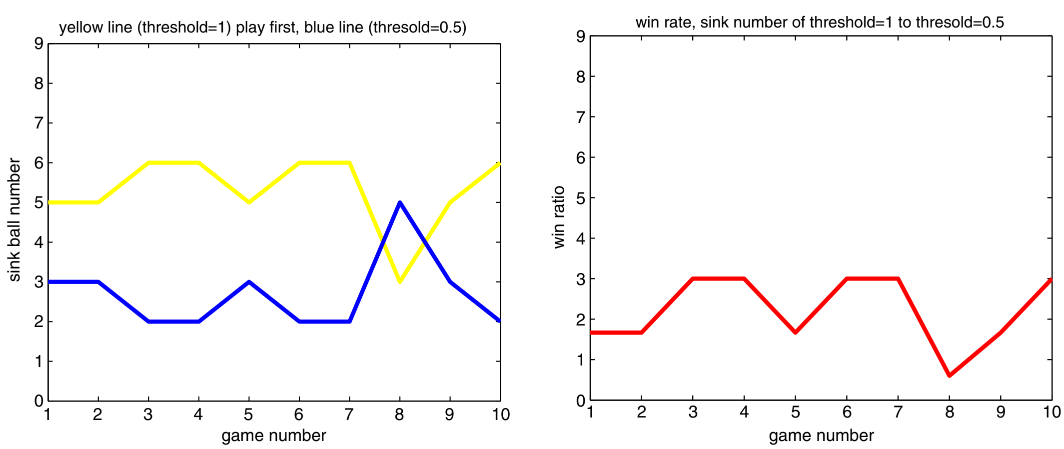


Fig. 12 Simulated ten games statistics of two players with similar threshold values (1 and 0.5) with 1 player playing first

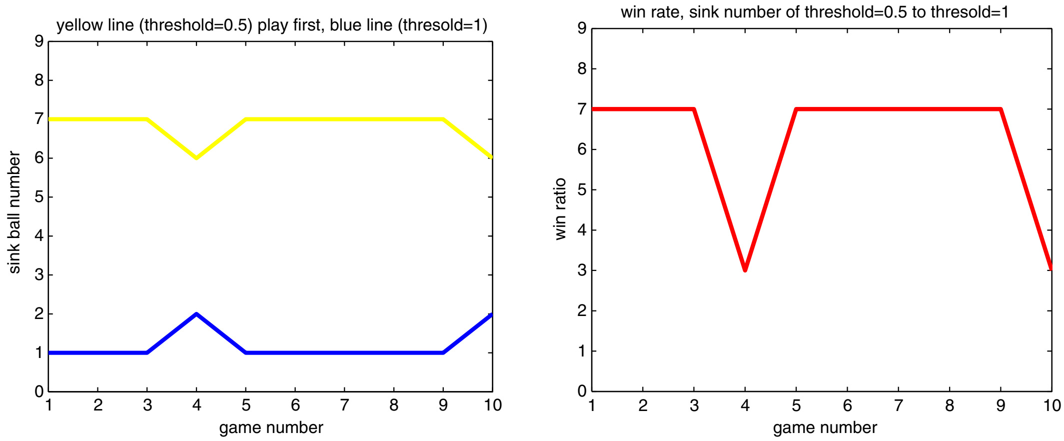


Fig. 13 Simulated ten games statistics of two players with similar threshold values (0.5 and 1) with 0.5 player playing first

the second player. The inadvertently missed shot of the second player further create opportunity for the first player to clear the table given the lower threshold indicating a higher capability to maintain consecutive shots.

More games are simulated per different threshold pairs in the competition. The total scored ball numbers are recorded at the end of a game for both players to compare the performance. Additionally, the win rate defined as the ratio of the total scored ball numbers between the first to the second player is calculated as performance comparison index. Figure 11 is the results of two players with high contrast threshold values of 1 and 0.0001. The player with threshold value 1 plays first. The results show that the first player scores the first three shots, and then lose the rest of the shots to the second player with very low threshold value for all ten games. The win ratio is calculated to be 0.6 from the first player's perspective. This phenomenon is not beyond expectation, as the second player represents a high proficiency player inherent with both good skills and strategy play. He can justify not only the optimal first shot but the subsequent shots after taking over the right of play. With carefully controlled handing on the cue stick, he can easily render tough shots and clear the table in one turn. This result supports the observation that when the proficiency of players differs in large extent, the order of play really doesn't influence the final outcome. In the other words, the system can not guarantee to minimize the potential loss of weak players given that the opponent has much better skills and strategy. This been said, the high skill player still uses the strategy proposed by this author, thus proving the effectiveness of our algorithm.

To further investigate the effect of player order on the gaming results, more games are simulated using the same threshold pairs as in Figs. 9 and 10. Ten games are exercised for each match. Figure 12 is the match results of two players with the threshold pairs of Fig. 9. Figure 13 is the match results of two players with the threshold pairs of Fig. 10. In Fig. 12, the weak player with the threshold of 1 plays first. From the results,

Table 1 Games statistics of higher skill player playing without guide

Game #	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10
High skill without guide	7	6	7	6	5	5	5	6	3	7
High skill without guide	1	2	1	2	3	3	3	2	5	1
Change hand frequency	7	4	7	4	4	4	4	4	8	7

Table 2 Comparison of games statistics of higher skill player playing with guide alternatively

Game #	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10
High skill without guide	3	3	3	3	3	3	3	3	3	3
High skill with guide	5	5	5	5	5	5	5	5	5	5
Change hand frequency	2	2	2	2	2	2	2	2	2	2

Table 3 Win rate comparisons of higher skill players among games using and not using author's repositioning guiding system based on Tables 1 and 2

Proficiency level	Maximum win rate	Average win rate	Max. performance enhancement	Avg. performance enhancement
Without guide	1.66	0.439	N/A	N/A
With guide	1.666	1.666	0	2.78

we can observe that the first player wins most of the game except in game 8 where he loses two balls to the second player. The average win rate is around 2.23 from the first player's perspective. This phenomenon affirms the observation that the order of the play does have impact on the outcomes of the competition given similar proficiency levels. In the case of Fig. 12, the lower proficiency player who plays first even outperforms the second player with slightly higher proficiency. The same phenomenon happens in Fig. 13 too with the lower threshold player playing first. The high proficiency player wins all of the games with high margin of scored balls over the second player with lower proficiency. The win rate is about 6.2. This value is about three times than that of Fig. 12. This phenomenon is also not a surprise and supports the fact that order of play does have impact on the final outcome. The high win rate is the compound effects of order of play and skill proficiency difference.

7.3 Competition Results Between Players with/without Guidance

Given the different driving speed (force), the players with different skill levels are arranged to test our guidance system integrated with the optimal reposition algorithm. Such experiments are done twice, one without velocity and aiming direction guide, the other one with such guide. Two set of players with different level of proficiency are called together for the testing. The first set of players is more experienced and the other is less experienced. The statistics of the first set experiments are included in Tables 1, 2 and 3. The second set's statistics are included in Tables 4, 5 and 6. Both Tables 1 and 4 are results of two groups of players playing without guidance system. From Tables 1 and 4, we can observe that the first player of the higher skill group won most of the game and the average win rate from the perspective of the second player is 0.439 from Table 3. This is an intuitively reasonable proof that this group of people does own higher skills. This is due to the advantage of the few beginning shots claimed by the first player. The same experiment is conducted for the second group of player with less experience. The statistics in Table 4 shows a contrast of scores to that of Table 1. The second player actually wins more games than the first player. The average win rate from the second player's perspective is around 1.382 from Table 6. This indicates that the proficiency level of this group of player is actually lower than the players in Table 1. The other evidence that this group is less experienced is the higher change hand rate from Table 4 (ranged from 5~9) compared to that of Table 1 (ranged from 4~7). Higher change hand rate means easy to miss shots and have to turn the right of play to the opponents. This phenomenon is quite typical for low skill players due to the awkward handling on the cue stick. So the experiments for playing without guidance system serve as a reference for their proficiency levels.

Table 4 Comparison of games statistics of lower skill player playing without guide

Game #	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10
Low skill without guide	5	3	3	3	3	4	3	5	3	3
Low skill without guide	3	5	5	5	5	4	5	3	5	5
Change hand frequency	8	5	9	9	7	5	5	5	9	9

Table 5 Comparison of games statistics of lower skill player playing with guide alternatively

Game #	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10
Low skill without guide	2	2	2	3	3	2	2	1	2	2
Low skill with guide	6	6	6	5	5	6	6	7	6	6
Change hand frequency	4	4	4	5	6	4	4	6	4	4

The second set of experiments arranges the same sets of players to use the integrated guidance system for testing its efficiency. The low skill group and high skill group all join the experiments. One player in each group plays without the help of guidance system, while the other plays with guidance system. Table 2 is the game statistics for higher skill players without and with the help of the guidance system. Table 5 is for lower skill players in the same condition. From the Table 2, the player using the guidance system outperforms the player without using the system in all ten games with an average win rate of 1.666 from the second player's perspective. In this setting, the player plays first without using the guidance, while the second player uses the system as assistance. What is worth noting is that the change hand rates for all games are all 2. This phenomenon indicates that our system can help enhance players' performance in rendering successful consecutive shots. With the help of our system in combination with better control of the cue stick, the second player actually manages to break the advantage of the first player in playing order.

Comparing results from Tables 1 and 2, Table 3 summarizes their differences in terms of the maximum win rate, the average win rate, the maximum performance enhancement and the average performance enhancement. The performance enhancement is defined as the ratio between the gains in win rate of using our system to the win rate without using our system. The maximum performance enhancement is zero, while the average performance enhancement of using our system is 2.78 for this set of players. This proves that our system can be helpful in assisting users in enhancing their skills and increase the fun of game.

From the Table 5, the player using the guidance system outperforms the player without using the system in all ten games with an average win rate of 3.12 from the second player's perspective. The maximum win rate is 7 for this set of experiment. In this setting, the player plays first without using the guidance, while the second player uses the system as assistance. The average win rate for this set of players is almost twice that of the higher skill player in Table 2. This could mean that the lower skill player actually relies on our system in winning the ten games in large margin. Although the missing rate is high as observed from the higher change hand rate than that of Table 2, our system does contribute to the final total sunk ball numbers.

Comparing results from Tables 4 and 5, Table 6 summarizes their differences in terms of the maximum win rate, the average win rate, the maximum performance enhancement and the average performance enhancement. The maximum performance enhancement is in the ratio of 7, while the average performance enhancement of using our system is 2.25 times higher than that without using our system for this set of players. The maximum win rate enhancement of this set of users shows a drastically high ratio, indicating that the low skill player benefits more than high skill player in using our integrated system. The average win rate enhancements for the both the high and low skill players are rather high and all beyond 2. This evidence provides even more solid proof about the reliability of our system in helping users in real world competition scenario.

8 Conclusion and Future Works

The selection of a best sequential shots for a given start cue position is a major challenging task in a billiard game. Additionally, the play order sometimes influences the final outcome within a competition. This work is the first to investigate

Table 6 Win rate comparisons of higher skill players among games using and not using author's repositioning guiding system based on Tables 4 and 5

Proficiency level	Maximum win rate	Average win rate	Max. performance enhancement	Avg. performance enhancement
Without guide	1.66	1.382	N/A	N/A
With guide	7	3.12	4.21	2.25

Table 7 Summaries of competition simulation results

Players proficiency (threshold value)	High skill player wins all	Play order decides
Low(1) to Extreme high (0.0001)	Yes	No
Low(1) to High(0.5)	No	Yes
High(0.5) to Low(1)	Yes	Yes(help increase win rate)
Low(1) to Low (1)	N/A	Yes
High(0.5) to High(0.5)	N/A	Yes(help increase win rate)

the effects of consecutive shots and order of play on the billiard gaming results. A new strategy is proposed to apply maximum tolerance angle search sequentially. One on the first shot and the second on the subsequent shot alone the path after colliding with a best selected object ball with maximum tolerance. The strategy considers combinations among all pockets and target object balls during both the pre and post collision shots selection processes. A simulation program is developed to test the strategy in a competition scenario. An algorithm allows users with different proficiency to apply this strategy in the real gaming situation. The order of computation complexity of the program is analyzed in the order of third degree polynomial. This is caused by the triple search operation of the nested loop on the post collision path for an optimal position best for follow up shots.

Table 7 summarizes the discoveries of the competition simulation program. The competition cases are divided into five categories. The first class is when the proficiency difference is large with the lower skill playing first. The results show that the extreme high skill play easily scores most of the balls and win the game. In this case, the order of play doesn't influence the gaming results at all. This is due to the capability of the relative high skill player in rendering the tough shots left by the low skill player. This is consistent with the real gaming situation where the skilled player usually out runs the low skill player in various balls configurations. This phenomenon indicates that the simulation program can predicate the regular competition results accurately. In the case of low to high proficiency, high skill player actually loses the game to the low skill player. The order of player does have impact on the competition results. With the help of our guidance system, the low skill player actually scores some more points in the few beginning shots. This lead in points lasts through out the game and help win the final competition. This result is again another proof of the effectiveness of our algorithm in helping players in improving their skills.

In the third case where the high skill player matches with the low skill player, the high skill player wins all the games. The order of play compounding the effective usage of our system contributes to the high win rate of the final competition. For the cases where the players have equal proficiency the play order decides who win the game. Generally, the one who plays first wins the game. The win rate is much higher for the high skill players. This can be attributed to the higher number of successive successful shots the high skill player enables.

A vision based guidance system is devised to test the proposed strategy. The system is modified from a previous design to instruct users for reliable shots aiming. A repositioning controls instruction is first analyzed then displayed without having to track the cue stick. The players with different skill levels are arranged to test our guidance system integrated with the optimal reposition algorithm. Such experiments are done twice, one without velocity and aiming direction guide, the other one with such guide. One set of players is more experienced and the other is less experienced. The two groups of players first play without guidance system. From Tables 1 and 4, we can observe that the first play group with higher skill won most of the game in large win rate score, while the first play group with lower skill fails to win all games. This is an intuitively reasonable proof of each group's skill levels.

From the Tables 2 and 5, the player using the guidance system outperforms the player without using the system in all ten games with an average win rate of 1.666 and 1.382 from the second player's perspective. What is worth noting is that the change hand rates for all games of high skill player are all 2. This phenomenon indicates that our system can help enhance players' performance in rendering successful consecutive shots. With the help of our system in combination with better control of the cue stick, the second player actually manages to break the advantage of the first player has in playing order. In all, the simulation program can provide an initial proof of the effectiveness of the reposition algorithm in the competition situation. The data from Section 7.3 further provides evidences that our vision tracking system, front end guiding interface, ball placement algorithm and deceleration rate calibration are fully integrated and constitute an efficient and precise training system as a whole.

The next level of research will focus on extending the search algorithm by considering the case of cue rebound and different object balls sinking orders instead of the maximum tolerance candidate. This will allow enlargement of the search space of the optimal cue relocations, and hopefully will lead to higher successive sink rate which is crucial for user to score high and win a game both in the simulation and real game world.

Acknowledgement The author wishes to thank the financial support of National Science Council of Taiwan for completion of this work under contract number of NSC 98-2220-E-029-003 and NSC 99-2220-E-029-003.

References

1. Jebara, T., Eyster, C., Weaver, I., Starner, T., Pentland, A.: 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)
2. Grogono, P.: Mathematics for Snooker simulation. Personal communication (2001)
3. Larsen, L.B., Jensen, P.M., Kammersgaard, K., Kromann, L.: The automated pool trainer—a multi modal system for learning the game of pool. In: Proceeding of the International Conference on Intelligent Multimedia and Distance Education (2001)
4. Jouaneh, M., Carnevale, P.: The development of an autonomous robotic system for playing mini-golf. IEEE Robot. Autom. Mag. **10**(2), 56–60 (2003)
5. Hashimoto, K., Noritsugu, T.: Modeling and control of robotic yoyo with visual feedback. In: Proceedings of the IEEE International Conference on Robotics and Automation 3, 2650–2655 (1996)
6. Nakai, H., Taniguchi, Y., Uenohara, M., Yoshimi, T., Ogawa, H., Ozaki, F., Oaki, J., Sato, H., Asari, Y., Maeda, K., Banba, H., Okada, T., Tatsuno, K., Tanaka, E., Vamaguchi, O., Tacyhimon, M.: Volleyball playing robot. In: Proceedings of the IEEE International Conference on Robotics and Automation 2, 1083–1089 (1998)
7. Hoffman, J., Malstrom, E.: Teaching a miniature robotic manipulator to play chess. Robotica **1**(4), 197–203 (1983)
8. Groen, F.C.A., den Boer, G.A., van Inge, A., Stam, R.: Chess playing robot. IEEE Trans. Instrum. Meas. **41**(6), 911–914 (1992)
9. Acosta, L., Rodrigo, J.J., Mendez, J.A., Marichal, G.N., Sigut, M.: Ping-pong player prototype: a pc-based, low-cost, ping-pong robot. IEEE Robot. Autom. Mag. **10**(4), 44–52 (2003)
10. Smith, M.: PickPocket: a computer billiards shark. Artif. Intell. **171**, 1069–1091 (2007)
11. Leckie, W., Greenspan, M.: Pool physics simulation by event prediction 1: motion transitions. International Computer Games Association Journal **28**(4), 214–222 (2005)
12. Leckie, W., Greenspan, M.: Pool physics simulation by event prediction 2: collisions. International Computer Games Association Journal **29**(1), 24–31 (2006)
13. Shih, C., Hsiung, P.A., Wan, C.H., Koong, C.S., Liu, T.K., Yang, Y., Lin, C.H., Chu, W.C.: Integration of a vision-based tracking platform, visual instruction, and error analysis models for an efficient billiard training system. Opt. Eng. **48**(2), 027202 (2009)
14. Shih, C.: Zero tolerance cue angle analysis and its effect on successive sink rate of a low cost billiard reposition control tutoring system. Knowl.-Based Syst. (2011). doi:10.1016/j.knosys.2011.05.008