Roboshark: a gantry pool player robot

Mohammad EBNE ALIAN <m_e_alian@mehr.sharif.edu>

Saeed BAGHERI SHOURAKI <SBagheri@ce.sharif.edu>

M.T. MANZURI SHALMANI <Manzuri@sina.sharif.edu>

Pooya KARIMIAN <karimian@ce.sharif.edu>

Payam SABZMEYDANI <payam_s@ce.sharif.edu>

Computer Eng. Department
Sharif University of Technology
Tehran, Iran

Abstract: *Robots are coming to conquer new domains of everyday life. They are used in fields such as education, entertainment and housekeeping more than ever. Roboshark, a gantry robot capable of playing pool games, can be used for both training amateur pool players and entertaining amateur or professional ones. Roboshark belongs to gantry robots family. It is located directly above the pool table – a* $90^{cm} \times 45^{cm}$ *model in our project - and has four degrees of freedom. The first three are prismatic, correspondent to movement along the X, Y and Z axis in a 3 dimensional Cartesian space, and the last is revolutional in order to rotate the kicker arm. For gathering the environmental information including the location and the color of the balls, a camera is mounted over the table. Having the information sent by the camera, a frame grabber captures each frame as a still image in memory. Due to the importance of ball colors, a color blob detection based algorithm is used. It contains some noise removing and merging filters, which are applied on the found objects. Roboshark uses crisp mathematics to compute the strength and the angle of the shot (named "shot vector" in this paper) and a fuzzy approach to select the best shot. Only direct shots are taken into account at this step. We will show that with 16 fuzzy rules, in most cases the ball selected by the fuzzy target ball selection module is the same as the one which an expert will choose in the same situation*

Key Words: *Robotics, Pool games, Machine Vision, Fuzzy rule base*

1-Introduction

For years, robots were designed to be employed in dangerous or exhaustive domains. The term "robot" - servant – shows the fact by itself that they were originally aimed to do the tasks which are unpleasant, difficult or impossible for human to do.

However, after years of experience and by having improved abilities and reduces costs, they are coming to conquer new domains of everyday life. Once it was quite a luxury to own a chess-playing or piano-playing machine, but today it only costs a few Dollars to buy a toy robot. They are used in fields such as education, entertainment and housekeeping more than ever.

Roboshark, a gantry robot capable of playing pool games, can be used for both training amateur pool players and entertaining amateur or professional ones. It is originally used to play the game snooker – one of the most popular games in Europe – but with some minor changes in software, it can be used for all kinds of pool games.

Snooker is known as the game of precision. Once a shot is chosen, the more precise kick you make, the better player you are, and the best player is the one who knows the best the exact point on the cue ball and the exact angle along with the exact power of the shot. The answer for a certain shot is unique and precise.

That is not the whole story anyway. Snooker is also the game of strategy. When you choose a certain ball to shot to a certain pocket, it is clear what to do next. But before that you have to select the most appropriate shot among all possible selections that is not such a clear deterministic problem.

Roboshark uses crisp mathematics to compute the exact point on the cue ball and the exact shot vector which is the direction and the strength of the shot. It was also possible to find a crisp model to find the most appropriate shot but it was neither simple nor humanistic. Instead, we use a fuzzy approach to select the best possible shot. It is necessary to mention that only direct shots are taken into account at the first step. Studies on angled shots (those in which at least one ball is reflected after hitting the bands of table) and indirect shots (those in which more than two balls are involved) are to be next.

2- Structure

Roboshark is a serial PC-based gantry robot. We have used a small $90^{cm} \times 45^{cm}$ pool table above which the axes of the robot are located. There is also a camera hanging exactly above the center of the table which provides the sensory input for the robot. Following we will describe the mechanical design and the logical function of the robot.

2-1- Physical Design

As a brief, Roboshark's mechanical body is a small $90^{cm} \times 45^{cm}$ pool table and an X-Y table above it. The X-Y table is equipped with two additional degrees of freedom. One is prismatic and changes the workspace of the robot from a plane (in X-Y table) to a bounded space. The other is revolutional which makes the end effector able to reach any point within its workspace with any horizontal orientation.

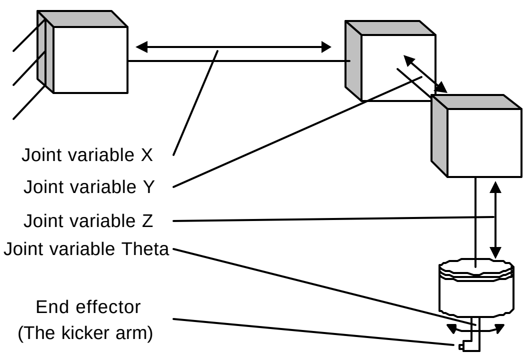

Figure 1: The schematic view of the robot's mechanism

The X-Y table is powered with 2 orthogonal stepper motors coupled to timing belts. Both the X and Y joint - variables have a precision of 01 mm and since this value is much further than the needed accuracy for pool games, we can ignore the little amount of existing backlash and use the open loop approach for the position control.

The joint-variable Z uses a vertical ball screw powered by a stepper motor, providing the precision of 0.02mm. Finally, the joint-variable ? rotates the end effector around the center of the cue ball using another stepper motor directly coupled to the end effector. The end effector is a pneumatic cylinder which hits the cue ball with a controllable strength.

Since the joint variables X, Y and the Z are prismatic and orthogonal, the inverse kinematic equations are solved with the minimum necessary computations. Figure 1 shows a schematic diagram of the mechanical design and table 1 gives some information about the physical design of the robot. Since in this paper we have focused mainly on the vision system and the shot selection's fuzzy rule base, we do not go for the mechanical details of the robot. For more information about on the details of mechanical design refer to [1].

2-2- Logical Design

There are four main modules which able the robot to play the game: Vision System Module, Fuzzy target Ball Selector Module, Crisp Computational Module, and Hardware Controller Module. An iteration of the game can be seen in figure 2.

The vision system and related algorithms are described in details in Section 3. In this module some features like the exact location of balls and packets are extracted from the picture provided by a camera mounted over the table. Based on these features, the important parameters for selecting the target ball and target pocket are computed.

Precision on the X axis	0.1 mm
Precision on the Y axis	0.1 mm
Precision on the Z axis	0.02 mm
Precision on the Theta axis	0.45 degree
Length of the X axis	750 mm
Length of the Y axis	1500 mm
Length of the Z axis	200 mm
Length of the Theta axis	360 degrees

Table 1: Some information about the robot mechanics

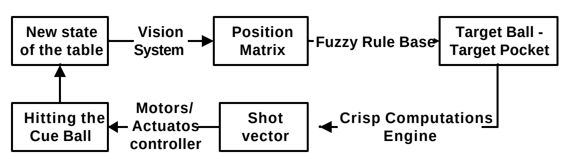

Figure 2: The schematic view of Logical Functions and Modules

A fuzzy approach is used to select the best possible target ball. By studying the pool games and knowledge acquisition from the experts, we could scratch the important parameters on which pool players select the target ball. These parameters (d_1, d_2 and q) are shown in Figure 3. In this figure, d_1 is the distance which the cue ball has to traverse before hitting the target ball, d_2 is the distanc e between the target ball and target pocket, and q is the angle between the two corresponding lines. A valid shot is a shot in which the following conditions hold:

$$d_3 < k \times r_b \quad (k > 2)$$

and

$$90 < q \le 180$$

Where r_b is the ball's radius (usually 2¼") and k is the safety factor close to and grater than 2. The parameter d_3 is the distance between the nearest ball other than cue ball or target ball to the lines D1 and D2.

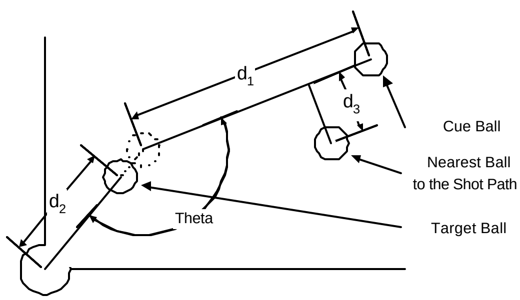

Figure 3: The parameters on which the best pair of ball-pocket is selected

Each possible shot denotes a ball-pocket pair. There is a total amount of $B \times P$ possible shoots, where B is the number of remaining valid balls on the table and P is the number of pockets around the table, usually equal to 6. For each possible shot if the above-mentioned conditions hold simultaneously, the shot is marked as valid and the parameters d_1 , d_2 and q are computed and send to the fuzzy rule base. The fuzzy rule based is described in details in section 4. It receives the parameters d_1, d_2 and q for all valid possible shots. It computes the most appropriate candidate and then passes the selected ball-pocket to the crisp computational module.

The crisp computational module receives the location of the target ball and the corresponding pocket from the fuzzy module and also the location of the cue ball from the vision module. It uses geometrical rules to compute the angle of the shot, and physical rules to compute the strength of the shot. These geometrical and physical rules can be found in [2] and [3]. The angle and the power of the shot are given to the hardware controller module to set the manipulators and actuators parameters and finally the shot is made. The angle and the power of the shot are taken to the hardware controller module to set the manipulators and actuators parameters and finally the shot is made.

3- Vision system

The only input from the environment is provided using a camera mounted over the field. This camera, an A131P Samsung, is adjusted so that the whole table is visible for it. The images will be captured by a frame grabber, Matrox Corona, and stored in a buffer in the computer's main memory. The needed information are ball colors and their positions. These positions should be as accurate as possible so the actuator controllers of the robot can do the desired job. To satisfy this need, an algorithm with the following specifications is designed and implemented. An overview of the vision system is also presented in figure 4.

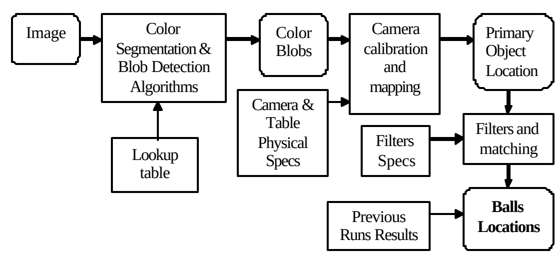

Figure 4: An overview of the Vision System

The entire algorithm for finding the objects is based on color segmentation, color blob detection, and object filtering and matching methods. The stored image in the memory is in RGB format and a lookup table is used to map any RGB vector to its correspondent ball color. Since the color is the most important feature in this method of segmentation, using color models other than RGB, like HSV and YUV, will result in a better color separation. Using a lookup table, as done in this project, will provide an excellent base to use any kind of color models or any other techniques like a combination of some color models. At the beginning of the program, the LUT will be filled using manually selected color ranges in the specified color model. For example, in this implementation every possible RGB vector is converted to its HSV equivalent vector first, and then by comparing this vector with particular color ranges for each ball color, the matching ball color will be stored in the LUT for that RGB vector. Hence during the object finding algorithm there is no need to conversion to other color models and the LUT is used for a direct conversion from RGB to the ball color number.

The main algorithm for finding objects is based on scanning the image line by line and making a list of objects and their features in it. A cursor is moved from top to bottom, line by line and forming some color blobs by adding the individual pixels to previously formed blobs. This addition will be made whenever a blob with the same color exists at the top or at the left of the current pixel. This action is done by updating the old same-colored blob's features considering the new pixel. In some cases, where both top and left pixels are the same color but belong to different blobs, like adding the new pixel, the two old blobs are merged too. After scanning the entire image once, a list of all color blobs along with their features is made. These blobs are our primary objects from which the balls will be extracted in the following parts.

The image seen by the camera is from a point above the table. By considering the optical features of the camera and its lens which causes a distortion, we can not have a linear and flat image. Also we need objects coordination in some standard length unit and not in pixels. To solve both these problems, a camera calibration algorithm is implemented. This algorithm partitions the table into four rectangles and uses a linear interpolation method to map each pixel of the image to its equivalent position in millimeters.

At this point, the found objects will be passed through some filters to find ball positions. These filters are adjusted considering the physical specification of the environment and frequent errors happened in vision system. The main stored features of the found objects are their color, number of pixels, bounding rectangle and center of mass. These features are used in different layers of the filters. In the first layer the noises which are known by their size and number of pixels are recognized and removed. The next layer is an object splitting layer. When two balls with the same color are so close that the main algorithm recognizes them as a single object, they should be separated according to our knowledge about ball sizes.

At the end of this part we will have the exact location of the objects which have passed our filters criteria. The only task remaining is to match these objects to our known balls. The whole above algorithm will run for several times, and the final result will be extracted from their average, so the environmental errors will be reduced as much as possible.

4- Fuzzy rule base

After Zadeh introduced his theory of fuzzy sets and fuzzy logic ([4] and [5]), his theory has been used for a wide range of problems. It is a perfect tool for solving problems including vague and uncertainty, or problems in which there is not a known exact model of the problem. It can also handle the linguistic parameters' values of the parameters and find a quite satisfying answer for the problem. For further study on linguistic synthesis of problems using fuzzy logic and fuzzy rule base refer to [6].

We decided to use a fuzzy rule base to select the most appropriate ball-pocket pair to make it simple and also similar the way human players select their target ball in the game. For this purpose, the parameters d_1, d_2 and q are passed to the fuzzy rule base. While 3 parameters are received by the fuzzy rule base, it uses only two (d_2 and q) for clustering the input space. Since the studies showed that d_1 is not as much important as the two other parameters, the input space is clustered only regarding to the parameters d_2 and q . This will result in a considerable decrease in the number of fuzzy rules. The affect of d_1 is taken into account when the output of the rule is computed.

The membership functions for the parameters d_2 and q are shown in figure 5. There are 4 membership functions for each parameter, resulting in 16 rules. These rules are shown in table 2.

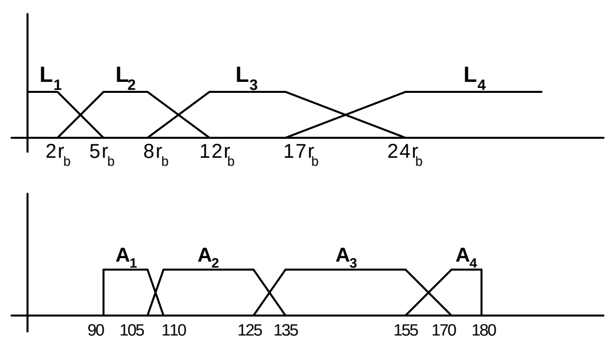

Figure 5: The membership functions for d_2 (up) and q (down)

The experiments showed that with these 16 rules, in most cases the target ball-pocket pair recommended by the fuzzy target ball selector is the same as the one selected by the expert players. We have to remind that only direct shots are taken into account at this step and also there is no comparison to the cases in which the player is planning for his/her next shot.

d_2	q	Output	d_2	q	Output
L_1	A_1	$\frac{1.30D-d_1}{D}$	L_1	A_3	$\frac{1.87D-d_1}{D}$
L_2	A_1	$\frac{1.20D-d_1}{D}$	L_2	A_3	$\frac{1.85D-d_1}{D}$
L_3	A_1	$\frac{1.15D-d_1}{D}$	L_3	A_3	$\frac{1.75D-d_1}{D}$
L_4	A_1	$\frac{1.10D-d_1}{D}$	L_4	A_3	$\frac{1.70D-d_1}{D}$
L_1	A_2	$\frac{1.65D-d_1}{D}$	L_1	A_4	$\frac{2.50D-d_1}{D}$
L_2	A_2	$\frac{1.60D-d_1}{D}$	L_2	A_4	$\frac{1.95D-d_1}{D}$
L_3	A_2	$\frac{1.55D-d_1}{D}$	L_3	A_4	$\frac{1.90D-d_1}{D}$
L_4	A_2	$\frac{1.50D-d_1}{D}$	L_4	A_4	$\frac{1.80D-d_1}{D}$

Table 2: The fuzzy Rule Base

The fuzzy rule base searches among all possible ball-pocket pairs. The recommendation of each pair is the sum of output values of each rule for that certain pair. The location of the recommended target ball and pocket are passed to the crisp computational module. This module receives these locations and also the location of the cue ball from the vision system and by using simple geometric rules, founds the angle and the kick point on the cue ball. The shot strength is also computed using the following equation:

$$P_s = {k.d_1.d_2}/{\cos q}$$

k is a coefficient depending on the physical characteristics of the table and balls, mainly the friction of the balls and the table's surface, and the coefficient of restitution for the balls.

These parameters are passed to the actuator controller module, where these parameters are translated to commands for stepper motors and pneumatic system. At this step the shot is made and another cycle of the whole algorithm begins.

5- Conclusion

This project combines a simple but accurate mechanism, a fast color-based approach for the vision system and a high performance fuzzy rule base to build a pool player robot. We are planning to employ some pattern recognition methods in the vision system to give the robot ability of playing other kind of pool games in which the balls are not totally solid, like 8-ball and 9-ball. The fuzzy rule base is the main point to be improved. We are working to define a critic for the rule base so that even if the initial rules are not fair, the robot "learns" to improve them and its playing skill subsequently. In the next version of the robot we will work on angled shots and indirect shots to give it more candidates to choose from. Also the strategy of the game and planning for the next shots are the issues we will face in the next step of the project.

References:

[1] M. Ebne Alian, "Designing and Implementing a pool player robot", Master Thesis, Computer department of Sharif University of Technology, 2003.

[2] W. C. Marlow, "The Physics of Pocket Billiards," ISBN: 0964537001, Philadelphia, PA: AIP, 1995.

[3] R. Shepard, "Amateur physics for the amateur pool player", 3rd Edition, 1997.

[4] Zadeh, L.A., "Fuzzy sets," Information and Control, Vol. 8, pp. 338-353, 1965.

[5] Zadeh, L.A., "Fuzzy Logic," Computer, Vol. 1, No. 4, pp. 83-93, 1988.

[6] Mamdani, E.H. and S. Assilian, "An experiment in linguistic synthesis with a fuzzy logic controller," International Journal of Man-Machine Studies, Vol. 7, No. 1, pp. 1-13, 1975.