Playing Pool with a Dual-Armed Robot

Thomas Nierhoff, Omiros Kourakos, Sandra Hirche

***Abstract*— This video presents a robot capable of playing pool on a normal sized pool table using two arms. For successfully completing this task several issues need to be addressed, including the perception of relevant environment information, planning of actions and finally an efficient execution. The video outlines how the robot accurately locates the pool table, the balls on the table and the cue and subsequently plans the next shot. In order to improve the stroke speed, an optimization algorithm for the arm configuration is described. Finally, it is shown how all these modules are integrated to achieve a working two-handed robotic pool play.**

I. INTRODUCTION

During the past two decades autonomous robotic systems for entertainment purposes have started to appear. One popular domain is pool robots. Presumably the first one was developed at Bristol University from the mid-80s until the mid-90s [1], [2]. The history of pool robots continued 2004 with Roboshark [3] and 2008 with Deep Green [4]. At the moment, Deep Green seems to be the most advanced one when it comes to playing ability. Based on a four DOF ceiling-mounted gantry robot, it is able to play at a "better-than-amateur level, with the ultimate goal of challenging a proficient human opponent at a championship level" [4]. Most of these systems have in common that they are mounted on the ceiling above the pool table without the ability to use a standard cue but rather a pneumatically driven stub [2], [4]. Very recently a different approach has been presented by WillowGarage [5]: Using their PR2 platform, the robot is able to move around the table and pocket balls using two hands with modified extensions for a proper cue hold and a normal cue.

Similar to the results presented by WillowGarage our aim in this work is to develop two-handed pool play with a standard cue on our robot. The four basic functionalities to be developed are:

- accurate localization of the pool table, balls and cue;
- deciding which ball to pocket next;
- navigation of the robot to an appropriate striking position;
- achieving the accuracy and dynamics required for a successful stroke.

As our work focuses on the first and last point, only they will be described more in detail together with some final results in the following section.

This work is the result of the Master thesis by Thomas Nierhoff. Omiros Kourakos and Sandra Hirche are with the Institute of Automatic Control Engineering (LSR), Faculty of Electrical Engineering, Technische Universität München, D-80290 München, Germany {tn, omiros.kourakos, hirche}@tum.de.

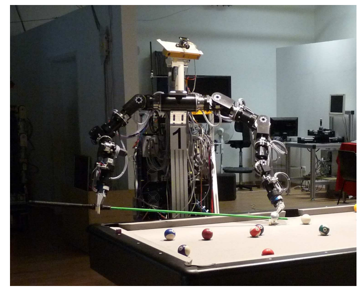

Fig. 1. Overview of the complete system

II. THE POOL PLAYING ROBOT

A. Hardware

The setup consists of a normal sized pool table with a camera with a resolution of 640x480 pixels mounted on the ceiling approximately 2.5m above the table. The robot consists of an omni-directional platform and two 7 degrees-of-freedom (DOF) manipulators mounted in a mirrored configuration with special endeffectors. Each endeffector consists of a two-axis gimbal set, which allows it to rotate freely while keeping the cue mounted on the innermost gimbal immobile. The left arm serves as support for the cue tip and also for an accurate positioning of the cue, while the right arm executes the stroke.

B. Perceiving the environment

Only the ceiling mounted camera and the arm joint position data are used to extract all the relevant information for playing pool. By using a model based approach, where the dimensions and the color of the borders and the play area of the pool table are known, the position of the table, all balls and the cue are determined. First a combination of background subtraction and histogram comparison based on the predominant color in an area around the center of the image for distinguishing between the table and balls on the image is used. Subsequently a Hough transform for lines and circles is applied to detect the table edges and the balls respectively. By merging information of a sequence of images, balls can be detected with subpixel accuracy (one pixel is about 4mm x 4mm on the table, a ball has a diameter of approx. 18 pixels).

To detect the cue, a combination of the cue's geometry and its previously learned color is used. The image is thresholded with respect to the cue color in the HSV color space. Next, a Hough transform for lines is performed. By taking only the longest lines into consideration, the approximate cue direction and position is detected. To refine detection, a

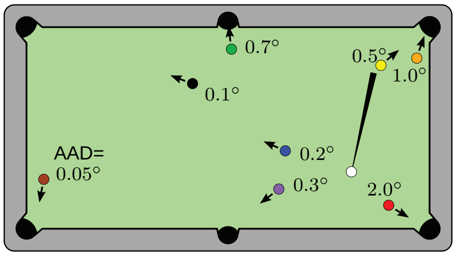

Fig. 2. Allowed angular deviation (AAD) in degrees for several balls and the corresponding pocket. The higher this value is, the easier the ball can be pocketed. The black circle segment emerging from the white ball shows the allowed deviation. The angle is 2·AAD - in this special case 1°

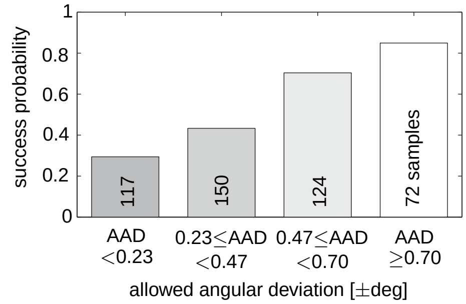

Fig. 3. Success probability to pocket a ball over allowed angular deviation (AAD) for a total of 463 strokes at various positions.

regression line along the cue direction based on the left and right boundary of the cue in the binarized image is calculated. Last, by extracting the endeffector height of both arms from the robot a 3D position of the cue is obtained. If the cue's exact position with respect to the table and the way the robot holds the cue is known, one is able to calculate the position of the robot with respect to the table.

C. Stroke Optimization

The 7 DOF design of the arms in addition to the gimbal based mounting system for the cue results in three redundant degrees of freedom. By taking advantage of the additional degrees of freedom the goal is to find an arm configuration that maximizes the achievable stroke speed. The optimization algorithm samples n points along the stroke trajectory, calculates the maximal achievable velocities $\dot{x}_{max}$ and accelerations $\ddot{x}_{max}$ for each point and maximizes the following cost function c, while keeping the joint velocity and torque limits

$$c = \prod_{i=1}^{n} \left(\|\dot{\mathbf{x}}_{i\ max}\| \cdot \|\ddot{\mathbf{x}}_{i\ max}\| \right). \tag{1}$$

The velocities in task space required for the optimization are coupled to the robot joint velocity limits through

$$\dot{\mathbf{x}} = \mathbf{J}\dot{\mathbf{q}}, \tag{2}$$

and the accelerations required for the optimization are coupled to the torque limits through the dynamic manipulability equation [6]

$$\ddot{\mathbf{x}} = \mathbf{J}\mathbf{M}^{-1} \cdot (\boldsymbol{\tau} - \mathbf{C} - \mathbf{g}) + \dot{\mathbf{J}}\dot{\mathbf{q}}. \tag{3}$$

The optimization is performed using the Nelder-Mead algorithm [7].

D. Results

In addition to the above described algorithms a simple path planner for the platform movement around the table has been integrated, which allows the appropriate positioning of the robot for executing a stroke. Moreover, a planner which computes the stroke difficulty is used to determine the best next stroke.

For performing each stroke, first the easiest shot is chosen. The allowed angular deviation (AAD) is used to determine the difficulty of each stroke. It denotes how much (in degrees) one can deviate from the optimal cue position still being able to pocket a ball without cushion contact (see Fig. 2 for a sample scenario). Subsequently for a given stroke, the optimization of the arm configuration is performed. Once the platform has reached the desired pose to execute the stroke, a first positioning of the cue on the table is performed. However, due to small position errors after the first positioning, fine positioning has to be performed in addition. Finally, the stroke is executed.

In order to evaluate the pocketing success rate an experiment with 463 executed strokes for randomly chosen ball configurations has been performed. The success probability for the easiest stroke class (AAD≥0.70°) is more than 80% and it almost linearly decreases with increasing stroke difficulty, see Fig. 3.

III. CONCLUSIONS

This video presents a two-armed robot capable of playing pool. The main focus in this work was on the perception of the pool environment and on the actuation of the robot to efficiently perform the strokes. The vision algorithms provide the accuracy required to successfully pocket the balls. In addition, the optimization of the arm configuration improves the dynamics of the robot.

ACKNOWLEDGEMENT

This work is supported in part within the DFG excellence research cluster *Cognition for Technical Systems - CoTeSys* (`www.cotesys.org`).

REFERENCES

[1] S. W. S. Chang, “Automating Skills using a Robot Snooker Player,” Ph.D. dissertation, Bristol University, 1994.
[2] T. M. K. H. L. Ho and J. Baldwin, “Snooker Robot Player - 20 Years on,” in *Proceedings of the 2007 IEEE Symposium on Computational Intelligence and Games*, 2007.
[3] M. E. A. et al., “Roboshark: A gantry Pool Player Robot,” in *35th International Symposium on Robotics*, 2004.
[4] M. G. et al., “Toward a Competitive Pool Playing Robot: Do we need Intelligence to play Pool?” *IEEE Symposium on Computational Intelligence and Games, Honolulu, Hawaii*, 1-5 April 2007.
[5] WillowGarage. The PR2 Plays Pool. http://www.willowgarage.com.
[6] M. T. Rosenstein and R. A. Grupen, “Velocity-dependent Dynamic Manipulability,” in *International IEEE Conference on Robotics and Automation*, 2002.
[7] J. A. Nelder and R. Mead, “A Simplex Method for Function Minimization,” 1965.