International Journal of Engineering Issues
Vol. 2015, no. 2, pp. 45-51
ISSN: 2458-651X
Copyright © Infinity Sciences

Automated Chess Playing with a Robot Manipulator

Dimitrija Angelkov, Natasa Koceska, Saso Koceski

Faculty of Computer Science, University Goce Delcev - Stip, Stip, Macedonia
Email: saso.koceski@ugd.edu.mk.

Abstract- This paper describes a system for automated chess playing with a robot manipulator. Customized chess engine is used to implement chess rules, to evaluate the board position during the game and to compute the next move of the robot using the alpha-beta search algorithm. This work contributes to the recent trends for creating automated robotic games and introduction of non-standard human-computer interfaces.

Keywords**:** *robot manipulator, computer vision, human – computer interaction, board games.*

I. INTRODUCTION

Board games have been very popular since ancient times and have been played in all cultures and societies. Main characteristic of all board games is that they are based on movement of various objects over a specifically designed board according to pre-defined set of rules. Most of the popular board games in the history are a sort of adaptation of a real battle between armies. Even contemporary board games are implementing the logic of defeating the opponent.

Learning curve strongly depends on the type of game and the complexity of its rules. However, some of the board games simple set of rules but, in order to master the game the player must develop profound strategies. One of these games is the chess game. Chess is a popular game among all ages, from young children to elderly adults. It is a game that exercise the brain and develops mental abilities as concentration, critical thinking, abstract reasoning, problem solving, pattern recognition, strategic planning, creativity etc.[1-6]. Playing chess includes various phases such as: perception of the board and game pieces, perception of the opponent, reasoning about the game state, and manipulation of the physical pieces while coordinating with the opponent.

Chess is a two-player board game played on a chessboard, a checkered game board with 64 squares arranged in an eight-by-eight grid. The two opponents can be human, or human and robot (computer). Advanced computer technology allows creating a computer games that mimic the human behavior. Such games are traditionally played with input devices like keyboard and mouse. Nowadays other means of interaction have been investigated that let user to focus on content rather than on input device [7-11].

Variety of chess automata developed through the history advocate that the robotic chess game could be very interesting from entertainment perspective. Historically, the Turk, was probably the first mechanical device aimed at chess playing automation. It was presented in the eighties of the eighteen century, as a clockwork machine that played chess. Although the real working mechanism is still not clear because the original Turk machine has being destroyed in a fire, according to various written reports from that times, one can conclude that it was not autonomous at all. The Turk used magnetically instrumented chess pieces that enabled the operator to sense the game state via the motion of corresponding magnets in the operator's compartment. Thus the mechanical Turk was actually a chess teleoperation system, not a chess automaton [12], [13].

Recently various robots that enable chess automation have been presented in the literature. Chess playing robot arms were shown in 2010 at the Maker Faire [15], and at several press events in Russia [14]. They used instrumented chess sets such as the Digital Game Technology Sensory Chess Board [16] to eliminate the perception problem. A commercial product called the Novag 2 Chess Computer [17] includes a robot arm that can play against a person. However, it uses a special instrumented board for perception and special chess pieces that are co-designed with the manipulator.

In this paper we present a low cost automated chess game played with a robot arm manipulator. It can play with chess pieces with an arbitrary form and on a variety chess boards. The proposed system uses a computer vision system to continuously monitor the chess board and track the movements on the board.

II. SYSTEM DESCRIPTION

Hardware configuration of the system consists of a chess board and pieces, a robot arm to move those pieces, a vision sensor to view the game board, and a personal computer. The robot arm has 6-DoF with a parallel jaw gripper and six servos for movement of the joints. The schematic model of the robot manipulator is presented in Figure 1.

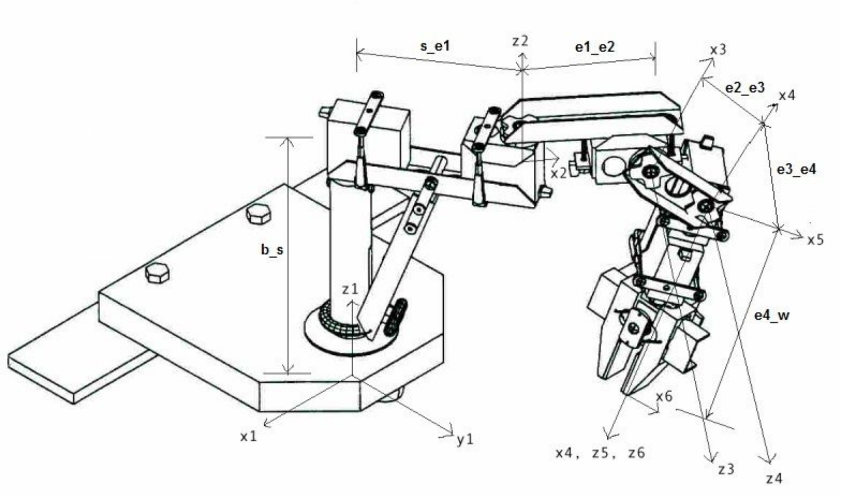


Figure 1. *Schematic model of the robot manipulator*

Dimensions of the links (between two consecutive connection points) are given in Table 1.

Table 1. *Dimensions of the links*

Link ID	Length (cm)
b_s	12
s_e1	10
e1_e2	10
e2_e3	12
e3_e4	12
e4_w	6

In our case we have to deal with the problem of inverse kinematics i.e. knowing the position and the orientation the end effector need to have (in order to pick up the chess piece) calculate the set of joint values to achieve this configuration. Generally we can define the problem of inverse kinematics as follows: Given a 4x4 homogenous transformation H, which represents the desired position and orientation of the end- effector

$$H = \begin{bmatrix} R & o \\ 0 & 1 \end{bmatrix} \in SE(3) \tag{1}$$

with R∈ SO(3) our task is to find (one or all) solutions of the equation:

$$T_n^0(q_1, q_2, ..., q_n) = H \tag{2}$$

Where $T_n^0(q_1, q_2, ..., q_n) = A_1(q_1) \cdots A_n(q_n)$ and q_1,..., q_n are the values for the joint variables that fulfill the previous equation.

The robot manipulator control algorithm is running on a PC which also runs the chess engine that plans the game moves. The real world image of the system configuration is given in Figure 2.

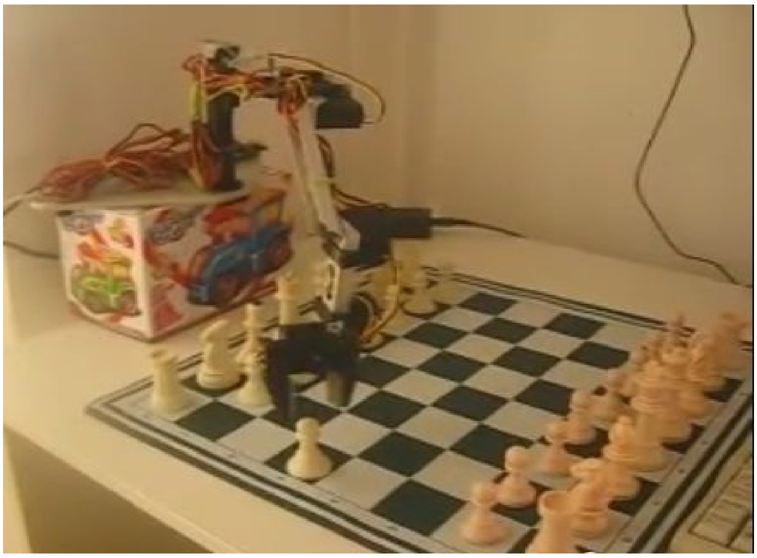
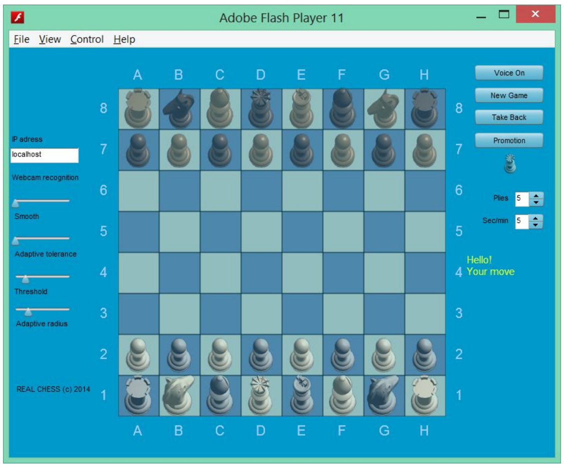


Figure 2. *Real world image of the developed system*

The idea behind the overall control algorithm is the following: the game starts with the player's move. The camera mounted on the top and connected via a standard USB port to a PC, capture the movement. Then the motion detection algorithm based on Canny edge-detection is used in order to detect the moves of the chessboard piece and to interpret the current positions of pieces on the chess board. The GUI of the developed application is shown in Figure 3.



IP adress: localhost

Plies 5

Sec/min 5

Hello!
Your move

Figure 3. *GUI of the developed application*

III. DESCRIPTION OF THE VISION ALGORITHM

The main input into the algorithm is provided by the web cam which is placed orthogonally to the chess board. It captures color images with resolution 640x480 pixels. The first problem with the captured images is that their brightness and contrast is depending on the environmental conditions. This issue is important because in the following steps consecutive frames should be compared to each other. In order to cope with this issue several solutions are possible such as: output transform (gamma curve), grayscale conversion or histogram equalization. In our case we have applied the grayscale conversion of the image. Afterwards algorithm foresees spatial transformations in order to correct the image distortion. In the next phase board segmentation is performed and afterwards, chess move detection algorithm is applied. In order to find out the direction of chess move the current frame is compared with the previous one without chess piece on it. Using edge detection, and a low threshold, it is very easy to locate the pieces and empty squares as shown on Figure 4.

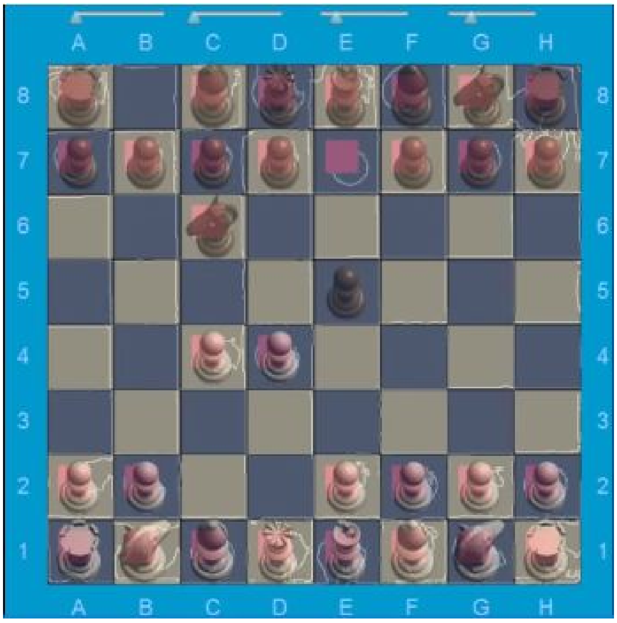


Figure 4. *Edge detection and thresholding algorithm*

IV. IMPLEMENTATION OF THE GAME LOGIC

Chess video game is implemented using existing chess engine. Chessboard is represented as an array with 64 elements. The movements of the pieces have to follow chess rules, such as “king can move one field in each direction” or “knight can move like L”.

The evaluation of the position is done based on the structure of the pawns and the occupation of the center. Computer generates moves that form a tree to a certain depth and searches for the best move. A common algorithm that is used for calculation of the best move is mini-max. In this algorithm each player tries to maximize his advantage and to minimize the advantage of the opponent.

To reduce the calculations during the search, alpha-beta pruning is used (Algorithm 1). The idea is to explore only useful variations, by fixing on each node a maximum or minimum evaluation. Those are computed by using the evaluations that have already been computed. In our case a search depth is set to a level 2, to speed up the calculation of the next move as an important characteristic of the interactive video games. At each step of the searching the function retains the best score it has found so far. When the maximizing player is searching a node and finds a path from that node having a higher score than the minimizing player’s best score, then he knows that the node from which he is currently searching is too good to be true since the minimizing player will never give him the opportunity to play the move to that node.

Algorithm 1: *Alpha-beta search algorithm*

```
function alpha_beta_search(real_side, search_depth, alpha, beta)
{
    var max_score = - 10000000; // initialization of best score
    var __reg3 = new Array();
    var catch_piece = null;
    var score_to_move;
    var moves_to_pc;
    var i = 0;
    while (i < 8)
    {
        var __reg1 = 0;
        while (__reg1 < 8){
            if ((chessBoard[i][__reg1] & 1) == side && chessBoard[i][__reg1] != VOID){
                __reg3 = __reg3.concat(moves_to_calculate (i, __reg1, side)); //concatenation of all moves
            }
                ++__reg1;
        }
        ++i;
    }
    var i = 0;
    while (i < __reg3.length){
        catch_piece = moving_this(__reg3[i], side);
        if (depth == 0){
            score_to_move = side == SIDE_COMPUTER ? pc_score : player_score;
        }
        else{
            score_to_move = 0 - alpha_beta_search(getOpponent(side), depth - 1, -1 * beta, -1 * alpha);
            // recursion of  alpha_beta_search function
            }
        unmoving_this(__reg3[i],side,catch_piece);
            if (score_to_move > max_score) max_score = score_to_move;
        if (max_score > alpha) alpha = max_score;
        if (alpha >= beta) return alpha;
            ++i;
    }

    return max_score;
    }
```

V. EVALUATION

In order to evaluate our system 10 test games were perform with a robot arm manipulator on one side and human on the other side. We analyzed problems encountered with perception, manipulation, and logic during these test games.

We have tracked how many success manipulation as well as failures occurred. For a failure manipulation the following cases we taken into account: failing to grasp a piece, dropping a piece, or failure to place a piece legally.

For a perception following cases were taken into account: failing to find a piece on the board (e.g., due to occlusion) or failing to detect illegal piece placement.

Results from the tests demonstrate the reliability of the proposed solution. Around 97% of all manipulations were successful, and only 3% were failures. Respect to perception errors only 1% failures occur during all the tests games. Logic errors (referring to failures to correctly identify an opponent's illegal moves) were not identified.

VI. CONCLUSION

This paper present a low cost automated chess game played with a robot arm manipulator. The proposed system uses a computer vision system to continuously monitor the chess board and track the movements on the board. The implemented motion controlled interface is able to recognize user's hand movements in front of the web camera and to move the selected piece according to that movement. The predetermined set of actions that the player can perform following the chess rules enables simple classification of the performed actions and interpretations of the movements. Customized chess engine is used to implement chess rules, to evaluate the board position during the game and to compute the next move of the robot using the alpha-beta search algorithm. This work contributes to the recent trend for creating games controlled by motion detection instead of standard computer input devices.

The tests show that the proposed system is highly reliable and robust. Because most of the errors encountered were manipulation errors, in a future we plan to improve the computer vision system by using some more information during the image processing phase.

REFERENCES

[1] CHARNESS, Neil. The impact of chess research on cognitive science.*Psychological research*, 1992, vol. 54, no 1, p. 4-9.

[2] CHARNESS, Neil, REINGOLD, Eyal M., POMPLUN, Marc, *et al.* The perceptual aspect of skilled performance in chess: Evidence from eye movements. *Memory & cognition*, 2001, vol. 29, no 8, p. 1146-1152.

[3] CHARNESS, Neil, TUFFIASH, Michael, et JASTRZEMBSKI, Tiffany. Motivation, emotion, and expert skill acquisition. *Motivation, Emotion, and Cognition: Integrative Perspectives on Intellectual Functioning and Development, Lawrence Erlbaum Associates, Estados Unidos*, 2004, p. 299-320.

[4] CHARNESS, Neil, TUFFIASH, Michael, KRAMPE, Ralf, *et al.* The role of deliberate practice in chess expertise. *Applied Cognitive Psychology*, 2005, vol. 19, no 2, p. 151-165.

[5] CHASE, William G. et SIMON, Herbert A. Perception in chess. *Cognitive psychology*, 1973, vol. 4, no 1, p. 55-81.

[6] CHI, Michelene TH. Knowledge structures and memory development.*Children's thinking: What develops*, 1978, vol. 1.

[7] HÄMÄLÄINEN, Perttu. Novel Applications of Real-Time Audiovisual Signal Processing Technology for Art and Sports Education and Entertainment.

[8] KOCESKI, Saso, KOCESKA, Natasa, et KRSTEV, Aleksandar. Hand Gesture–Based Control of Mobile Robot's Freight Ramp. 2010.

[9] SERAFIMOV, Kire, ANGELKOV, Dimitrija, KOCESKA, Natasa, *et al.* Using mobile-phone accelerometer for gestural control of soccer robots. In :*Embedded Computing (MECO), 2012 Mediterranean Conference on*. IEEE, 2012. p. 140-143.

[10] KOCESKI, Saso, KOCESKA, Natasa, et KOCEV, Ivica. Design and Evaluation of a Cell Phone Pointing Interface for Interaction with Large Projector based Displays. *International Journal of Computer Applications*, 2012, vol. 51, no 3, p. 27-32.

[11] PANOV, S., KOCESKI, S., KOCESKA, N. Usability aspects of eye gaze tracking systems. *Emerging Technologies for Better Living - ICT Innovations 2015*, October 2015, Ohrid, Macedonia.

[12] SCHAFFER, Simon. *Enlightened Automata*, in The Sciences in Enlightened Europe. University of Chicago Press, 1999.

[13] BURTON, H. The Turk: The life and times of the famous eighteenth-century chess-playing machine. 2002.

[14] BRATERSKY, A. Dvorkovich, Chess Robot Go 1-1. Available at:

http://www. themoscowtimes.com/news/article/dvorkovich-chess-robot-go-1-1/ 408089.html (December 2015).

[15] http://www.chessplayingrobot.com (December 2015).

[16] http://www.digitalgametechnology.com (December 2015).

[17] http://www.newyorkchessandgameshop.com/novag2robotchesscomputer.aspx (December 2015).