Autonomous Chess Playing System

Rituram Ojha
Department of Electronics and Computer Engineering, Thapathali Campus, Institute of Engineering, Tribhuvan University, Kathmandu, Nepal

Corresponding Email: tha076bei024@tcioe.edu.np

***Abstract*— With the increasing adoption of machine learning worldwide, Python is a popular programming language among machine learning engineers because of its simplicity and readability. This paper proposed the integration of python chess engine and a simple 2-DOF (degree of freedom) robotic serial manipulator to create a real time autonomous chess playing system. The moves from the human player is detected using overhead camera. The link is established between python chess engine and the robotic manipulator via serial communication using python API module (PySerial).**

Keywords: Computer Vision, Chess Playing Robot, Arduino, PySerial

1. INTRODUCTION

Chess is a popular strategy board game that is played between two players on an 8x8 board. It is believed that playing chess increases strategic ability a person. So, even after decades of its origin, chess still retains its popularity. The progress in the field of technology has given rise to the computer chess games and online chess hubs. But they don't give the feeling of playing real game of chess with chess boards and pieces. The motivation behind this autonomous chess playing system is to implement today's technology to make the game of chess more interactive and interesting with low cost and easily available resources. Chess being one of the popular strategy game requires more practice and training. Chess training centers can use this autonomous system to provide remote training sessions to children. Furthermore, The robotic manipulator is not limited to only playing chess but it can also find its application in industries and can be controlled effectively using its own set of control commands.

2. RELATED WORKS

Different methods has been attempted previously for the automation of chess game with specifically instrumented chess boards and pieces. Most common method is the use of overhead camera for the real time analysis of the chess board along with a robotic system for the movement of chess pieces on board. Prabin *et.al.*[1] proposed a system in which user's physical moves are fed to the system using a Logitech C270 overhead camera which takes real-time images of the board. Three copies of each frame are required for the three different tasks consisting up color segmentation for chess piece detection, edge detection for square center tracking and hand detection for process control. A computer numeric control (CNC) controlled magnetic moving mechanism is implemented for moving the chess pieces on board. Whereas Nandan *et.al*[2] proposed a system where they used similar algorithms for

piece detection but a simple 3-DOF (degree of freedom) robotic serial manipulator which is capable of playing chess in real time against any opponent is used.
The proposed system uses a overhead camera for the efficient detection of white pieces only using Color Filtering technique and a simple 2-DOF robotic serial manipulator for piece's movement on board. It eliminates the use of edge detection algorithm as well as hand detection from the system.

3. METHODOLOGY

This section discusses the methods employed for the design and implementation of the system. The system architecture of the proposed system can be seen in Fig. 1.

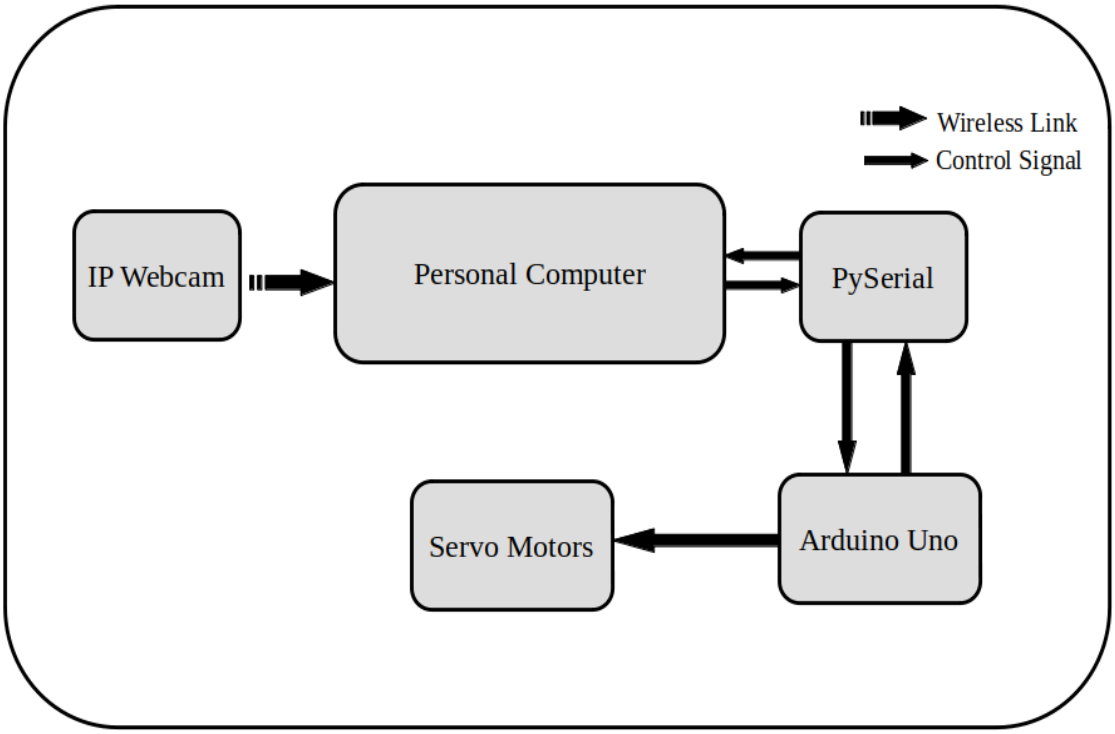

Figure 1 : System Block Diagram

The system consists of a camera directly above the chess board, a personal computer for image processing and chess engine, PySerial[3] module for serial communication between python chess engine and Arduino Uno and Servo motors for robotic manipulator. The real time status of the chess board is fed to the image processing system by webcam wirelessly. Then the white piece is filtered from the image and its movement is tracked throughout the game. The data flow diagram of the system is presented in Fig. 2. The proposed system architecture is explained below:

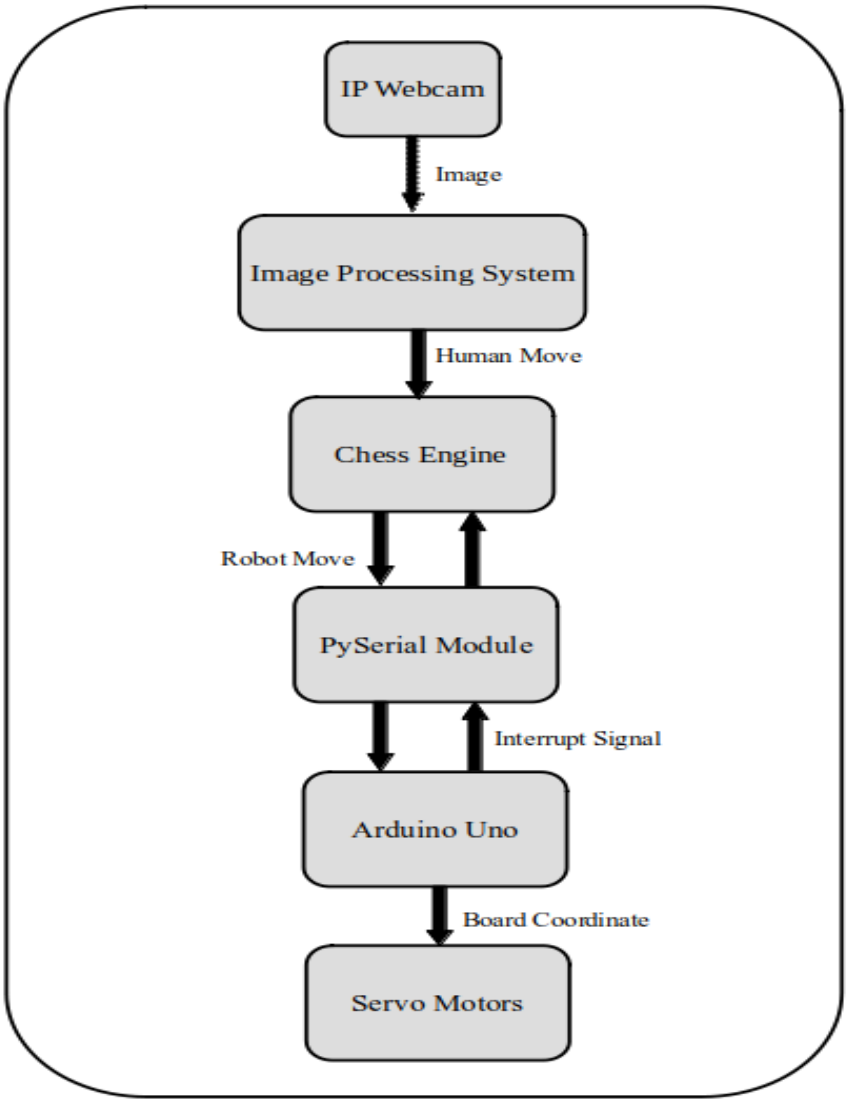

Figure 2: Data Flow Diagram

3.1. CHESS PIECE DETECTION

A overhead camera is positioned and the resolution was set at 1920 * 1080. The openCV image processing library[4] is used to capture the current frame and apply the image processing procedures. Lighting conditions have a major effect on the performance of the image processing techniques. The algorithm for this process works in the following steps:

A. Cropping the Image

The image obtained is cropped to select only the region of interest (ROI) that is the chess board as shown in Fig. 3. This removes unnecessary noise from the image and makes it easier to apply color filtering technique[5].

B. Color Filtering

The cropped image is converted from BGR color space to HSV color space. Since, the human counterpart is assigned to white pieces, we only need to detect the white pieces on the board. The lower and higher ranges of HSV values for white pieces is estimated. Then the image is masked with respect to the lower and higher ranges of values to obtain the binary image highlighting the position of white pieces in the board. The result is shown in Fig. 4.

Figure 3 : Region of Interest

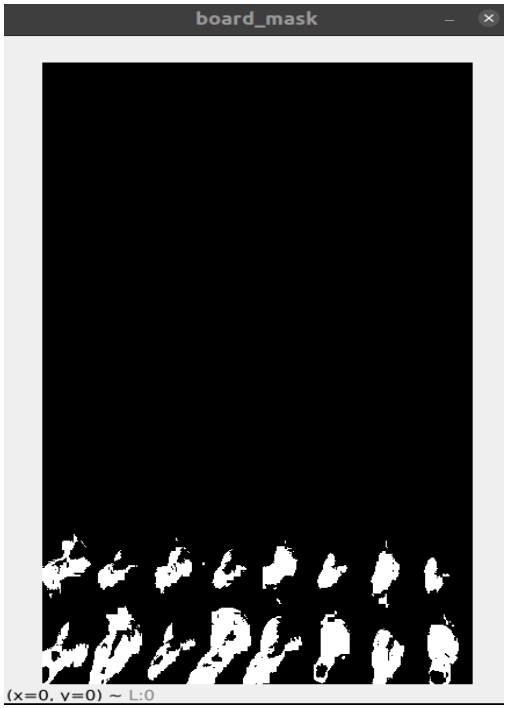

Figure 4 : Binary Image

3.2. PIECE MOVEMENT DETECTION

A. Square Occupancy Test

Here, The height and width of the obtained binary image is divided into 8 equal parts which provides altogether 64 images corresponding to each small squares of the board. Then, the number of white pixels of each images is compared with the threshold value to find out the square occupancy by the white piece.

Figure 5 : Piece on a board

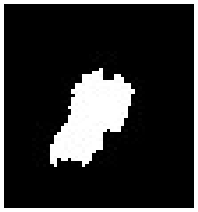

Figure 6 : Binary image of that piece

B. Binary Matrix Operation

Now, two 8*8 array is initialized to store the value corresponding to the previous and current status of the chess board. If any small square of the board is occupied by the white piece, the value '1' is assigned to the corresponding position of the array, otherwise '0'. At first, Both arrays are initialized as shown in fig. 7.

8	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0
2	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1
	A	**B**	**C**	**D**	**E**	**F**	**G**	**H**

Figure 7 : 8*8 array w.r.t chess board

After the human player have made a move, the new array is updated according to the current board status. The movement of piece from b1 to c3 is shown in Fig. 8. Now, the previous array and new array is compared such that the difference in the value of a particular index provides the latest move on the board by the human counterpart. That is, If any value is different and it is '0' on the new array, then it is regarded as starting location of latest move and if the value is '1' on the new array, it is regarded as final location of the latest move.

Figure 8 : piece moved from b1 to c3

8	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0
3	0	0	1	0	0	0	0	0
2	1	1	1	1	1	1	1	1
1	1	0	1	1	1	1	1	1
	A	B	C	D	E	F	G	H

Figure 9 : updated new array

3.3. CHESS ENGINE

It is written in python programming language as it is popular for its readability and less complexity. This chess engine is assisted by the python library called 'PyGame'[6] which allows you to create fully featured games and multimedia programs in the python language. The output from the image processing system i.e (Human Move) is fed as the input for the chess engine. Chess Engine provides the corresponding best move as the output. Now, Our robotic manipulator should perform according to this output. So, The output is further provided to Arduino Uno via serial communication with the help of python module PySerial. The corresponding move from chess engine to the human player's move is shown in Fig. 10.

Figure 10 : Game Window

3.4. INVERSE KINEMATICS

Inverse kinematics tells us the joint angles that we need in order to achieve a particular robot end effector pose.[7] As an example, To perform automated ice-cream serving, the robotic arm used needs precise motion from an initial position to the desired position between customers and ice cream container. In this process, the robotic arm obtain a specific value for the joints angle automatically according to the desired location of its end effector. In the Fig. 11 , The joints angles are given as q1 and q2, arm lengths as a1 and a2 and the location for the end effector is (x,y) with respect to world coordinate frame. Here, The joints angle q1 and q2 are determined geometrically.

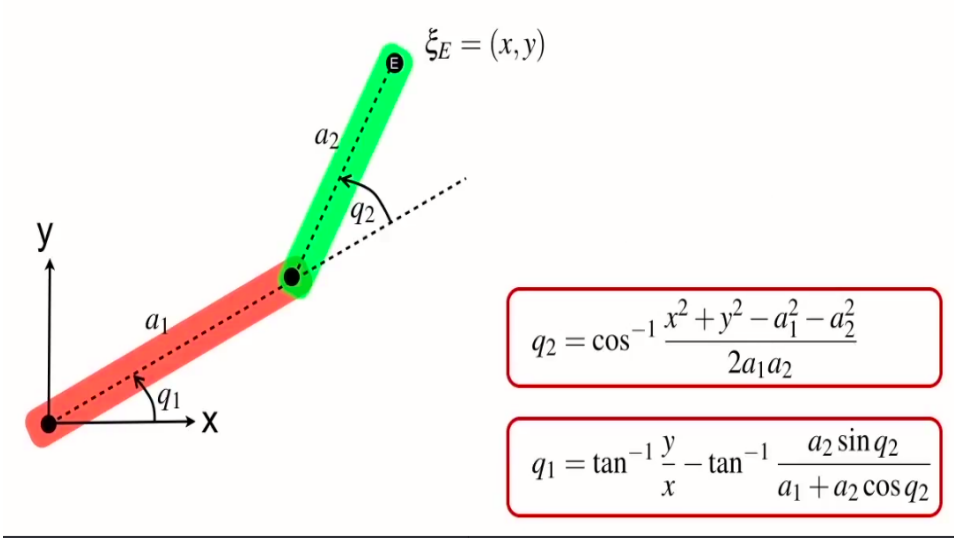

Figure 11 : 2-DOF robotic arm

The simple 2-DOF robotic serial manipulator used in our system also incorporate the idea of inverse kinematics for the 2 joint robot arm using geometry. The output of the chess engine is in the form of standalone positive integer. It is converted into coordinate value (x,y) assuming the lower right corner of the chess board as origin making the whole board as world coordinate frame as shown in Fig. 12.

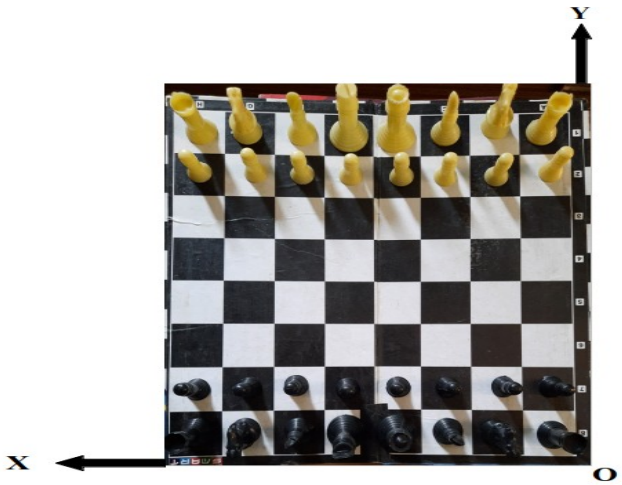

Figure 12 : Chess board as world coordinate frame

4. RESULT AND CONCLUSION

This system is designed to provide user a realistic environment while playing chess with the computer. The challenge was to create the system that works as a interface between physical world and chess engine. Lighting conditions on which the game is played has a major role in determining how efficient the system works. Sometimes, the shadow of the pieces due to irregular lighting are also detected as a piece by the system which causes error. So, a regular lighting environment provides best result. Also, The offset angles of servo motors are a bit of concern.

This system can be enhanced by the use of WiFi module for transmitting data from Chess engine to Arduino wirelessly. Further improvement can be done in making the system efficient in any lighting conditions using machine learning models.

REFERENCES

[1] P. K. Rath, N. Mahapatro, P. Nath, and R. Dash, "Autonomous Chess Playing Robot," *2019 28th IEEE Int. Conf. Robot Hum. Interact. Commun. RO-MAN 2019*, 2019, doi: 10.1109/RO-MAN46459.2019.8956389.

[2] N. Banerjee, D. Saha, A. Singh, and G. Sanyal, "A Simple Autonomous Robotic Manipulator for playing Chess against any opponent in Real Time," *Int. Conf. Comput. Vis. Robot.*, 2012, [Online]. Available: http://nandanbanerjee.com/files/ICCVR-08AUG12-011 paper.pdf

[3] "Welcome to pySerial's documentation — pySerial 3.0 documentation." https://pythonhosted.org/pyserial/ (accessed May 01, 2022).

[4] "Opencv - freeCodeCamp.org." https://www.freecodecamp.org/news/tag/opencv/ (accessed May 01, 2022).

[5] "Python Programming Tutorials | Color Filtering." https://pythonprogramming.net/color-filter-python-opencv-tutorial/ (accessed May 01, 2022).

[6] "Pygame Intro — pygame v2.1.1 documentation." https://www.pygame.org/docs/tut/PygameIntro.html (accessed May 01, 2022).

[7] "Inverse Kinematics for a 2-Joint Robot Arm Using Geometry | Robot Academy." https://robotacademy.net.au/?s=inverse+kinematics (accessed May 01, 2022).