**CHAPTER 1**

**INTRODUCTION**

Chess is a game in which two players separated by chessboard challenge the mental capacity of each other. The chess game starts with 20 possible moves and about 400 possible moves after first move each and this number exponentially increases with each move. After a few moves the possible moves becomes so complex that neither can imagine the all possible moves. No one thought that one day fast speed hardware clocks and sophisticated software will be able to solve such complexity level and even surpass the human intellectual capability.

Developing a chess program is considered to be the first discrete problem in the field of artificial intelligence and computation. This problem has discrete rules and clear goal (capturing King). The computer handles this complexity as tree problem. The root of the tree describes the current position of the game and each branch of the tree leads to the next possible legal move. It was assumed that if we solve the chess problem computationally, then we would be able to solve any other problem too. The strategy to find the best move (best tree) is called minimax. The minimax strategy is used with alpha-beta pruning which prevents minimax from going into branches that cannot produce better result than previous branches have already produced.

First most, an American mathematician Claude Shannon proposed computing routine for general purpose computer which was theoretical algorithm of chess program. In 1957, a Mathematician and IBM employee wrote first complete computer decode-able chess program which ran on IBM 704. It took about 40 years to build a computer chess program which could beat chess champions. In 1997, IBM created first best and brightest chess playing engine called Deep Blue. It was the first chess program that beat the Chess Champion Garry Kasparov in a 6 game match held in New York.

The chapter of autonomous chess playing robot was started in 1770, when a Hungarian engineer, Baron Wolfgang presented first so called chess playing robot, “The Turk” or “Chess Playing Automan”. The Turk had privilege to beat the statesman of that time including Napoleon Bonaparte and Benjamin Franklin. The Turk remained centre of fascination for 86 years until its destruction by fire in Chinese museum in 1854.

Several serious efforts have been made in the last decade to realize chess playing robot on industrial and academic level. Some used the magnetic sensor beneath each box to detect move and while some used digital chessboards. Such techniques exclude the image processing part and decrease the complexity level of whole project. The current work is an effort to develop a robotic system which can play board games. In this work, chess game was focused to challenge the dignity of human brain.

**CHAPTER 2**

**LITERATURE REVIEW**

*Hafiz Muhammad Luqman, Mubeen Zaffar, “Chess Brain and Autonomous Chess Playing Robotic System” in 2016* - A 4 degree-of-freedom (DOF) chess playing robotic manipulator and computer vision based chess recognition system are presented. The robotic system is capable of playing chess game autonomously against human or another robotic system. The logical system consists of anti-glare camera mounted on the chessboard, which acts as an eye of robot, personal computer for run-time implementation of computer vision algorithm, an open-source chess engine used as chess brain of robotic manipulator, which plays chess algorithmically on behalf of robot, and of course robot itself. On development side, a simple image segmentation based computer vision algorithm was developed to find the legitimate chess move and human hand motion detection, a software system was developed that enables the robot to pick and drop chess pieces from prescribed chess boxes, lastly communication channel was exploited for interfacing computer vision algorithm with chess engine. The whole robotic system is formulated from recycled machine parts and open source tools. [1]

*Cynthia Matuszek, Brian Mayton, Roberto Aimi, Marc Peter Deisenroth, Liefeng Bo,*

*Robert Chu, Mike Kung, Louis LeGrand, Joshua R. Smith, Dieter Fox, “Gambit: An Autonomous Chess-Playing Robotic System” in 2011 -* The Gambit, a custom, mid-cost 6-DoF robot manipulator system that can play physical board games against human opponents in non-idealized environments. Historically, unconstrained robotic manipulation in board games has often proven to be more challenging than the underlying game reasoning, making it an ideal testbed for small-scale manipulation. The Gambit system includes a low-cost Kinectstyle visual sensor, a custom manipulator, and state-of-the-art learning algorithms for automatic detection and recognition of the board and objects on it. As a use-case, we describe playing chess quickly and accurately with arbitrary, uninstrumented boards and pieces, demonstrating that Gambit’s engineering and design represent a new state-of-the-art in fast, robust tabletop manipulation. [2]

*Emir Sokic, Melita Ahic-Djokic, “Simple Computer Vision System for Chess Playing Robot Manipulator as a Project-based Learning Example” in 2008* – The proposed presents an example of project-based-learning (PBL) in an undergraduate course on Image Processing. The design of simple, low-cost computer vision system for implementation on chess-playing robot system is discussed. The system is based on stand CCD camera and a personal computer. This project is good tool to learning most of the course material that would otherwise be mastered by homework problems and study before an exam. An algorithm which detects a chess move is proposed. It compares two or more frames captured before, during and after a played chess move, and finds the difference between them, which are used to define a played chess move. Further image processing is required to eliminate false readings, recognize direction of chess moves, and eliminate image distortion. [3]

**CHAPTER 3**

**PROBLEM STATEMENT**

There was a chess playing robotic system introduced in 2011 known as “Gambit”, a robot manipulator system that was designed to autonomously play board games against human (or robotic) opponents. The Gambit system also introduces a custom robotic arm hardware which provides flexibility and accuracy over the chess board. Gambit can play with arbitrary chess sets on a variety of boards, requiring no instrumentation or modelling of pieces. Gambit monitors the board state continuously and detects when and what kind of move an opponent has made. Furthermore, Gambit communicates with a human opponent through a natural spoken-language interface.

The biggest drawback of the Gambit system is, both software and hardware do process very slowly. A chess program based on minimax strategy which find best move along with all possible moves. Because of this approach it consumes more time that ultimately increases process time. Same with robotic arm system, it takes more time to process and execute the operation. Plus the Gambit system was very noisy and it’s unable to recognize the height of chess pieces, so all the chess pieces must have same height.

To overcome these drawbacks, improvement has been introduced here. Improvements like upgrade in minimax strategy used with alpha-beta pruning which gives better result in less time. Also there is an improvement in hardware which processes operation faster and implement it. The noise control and chess piece recognition also improved.

**CHAPTER 4**

**PROPOSED WORK**

**4.1 IMAGE PROCESSING PART:**

A real-time computer vision algorithm is proposed for chess moves recognition and hand motion detection. The series of basic image processing operations are performed to accomplish the desired goal. The consistent chessboard design simplifies the task. It does not use complex or computational expensive visual detection algorithms. Instead of using complex and expensive algorithm, it uses simple computer vision algorithm. Firstly, a background registration technique is used to construct a base for background subtraction in later stages and its position will also be used as coordinate system for positioning of moving pieces. Then the moving object regions are separated by implementing change detection based segmentation algorithm which yields moving objects. Hand motion detection is implemented by computing the absolute difference of two consecutive images to measure the interval of move making process by human. Eventually, a post-processing step is applied on the object image to remove noise regions.

*4.1.1 Camera Setup:*

A web cam is used to keep the optimal quality. It provides the best quality images in low light conditions and avoids reflections. The pixel resolution of video is 640x480 in RGB format. The camera is mounted on chessboard at height about 18 inches to visualize the board from top-view. The frame is cropped to see only board, it saves our computation power and to avoid processing extra detected objects. The color-map of frames is converted from RGB to grayscale so that basic image processing operation can be applied.

*4.1.2 Image Distortion:*

The image is a 2D digital signal whose values are arranged in a matrix. Each value (pixel) gives information about the environment it captured. In practical imaging, some factors alter these values or distort the image which cannot be avoided and have to be encountered. The initial experiments were performed on animated images which gave ideal results. As animated images do not have any kind of noise, contrast imbalance, low intensity or distortion. When the actual images were captured in real environment, all these factors become bottleneck for attaining successful results. Octave is very handy tool which helped to implement the complex operations with a single command.

*4.1.3 Image Enhancement:*

In chess game, the chess pieces are positioned close to each other that cause shadows in the neighboring boxes. A pre-processing gradient filter is applied on the input frame to reduce the shadow effects. Sometime, standing people around the board occluded the light from their side which makes brightness of board non-uniform and also affects the contrast which is critical for the detection of chess pieces on the board. Image contrast enhancement is a crucial problem in image processing. There are several image processing techniques for the contrast enhancement in Octave (histogram equalization, histogram stretching, image sharpening and image intensity adjustment) which can be applied. In our case, image adjustment function worked best. It changes the intensity values of input grayscale image in such a way that only 1% values saturated around high and low intensities of the real image

*4.1.4 Chessboard Detection:*

The first step in chess game analysis is to detect the chessboard. The edge detection based segmentation technique is used to segment the whole chessboard into 65 segments. This segmented frame will be used in background subtraction in later stages. It makes the system computation faster and accurate. The colors selection of boxes was made on the basis of their grayscale values in such a way that, they give maximum contrast. It helps us to extract edge information. Conventional White and Black colors were chosen for boxes and Gray color for outside region of chessboard. In later case, the black chess pieces were marked with different color on the head to make them detectable on black boxes. The first step before placing the pieces on the board is to extract the position of corners of each box on the board and of the board itself and assign a label to each box (65 segments). The canny edge detection algorithm is used to detect boxes boundaries and pieces following by dilation operation. Now the camera position should be fixed until the end of game.

*4.1.5 Move Detection:*

When the initial process of setting a game has been completed and the camera has saved the background of the board. It sets to wait mode until hand motion is detected. As soon as it detects the motion in the video stream, it is set to alert state and keeps on checking the video frames unless motion is vanished. Now it captures another image and subtracts it from the image that was saved before motion and calculates the resultant image which shows change in the game situation. The detected piece coordinates are calculated and examined in which box these pieces lie. After determination of box number, chess move was assigned according to Algebraic Chess Notation. As the piece location was found by its position coordinates so the pieces should be placed in the center of box. Otherwise, it will not be able to find the box accurately.

4.2 CHESS ENGINE INTERFACE:

Chess Engine is a piece of software which is capable of battling against humans on chessboard. It is a beautiful creation of Artificial Intelligence. Fundamentally, chess engine does not come with GUI. It comes as a console application and a separate GUI (Winboard/Xboard) is interfaced to play interactively. Chess engine and GUI application communicate according to a protocol. The latest protocol is Universal Chess Interface*,* which is being used in this research. Chess moves are described, sent and saved in Algebraic Chess Notation. It is the standard chess notation of FIDE. The columns are named from a to h and rows are named from 1 to 8. Each piece is named by uppercase letter of its name such as King by **K** and Queen by **Q.** Every move has own its specific representation. Chess Engine sends moves in Algebraic Chess Notation to GUI applications. They interpret text and exhibit it in the form of graphical changes. In this case, the chess engine output move was the input of robot controller software. A communication channel had to be developed between chess engine and controller software. For this interfacing problem, the pseudo-chess engine logic was used developed. A fake chess engine was used to extract chess engine output in the form of algebraic chess notation. It works through file handling. Image processing algorithm writes its moves in the input file for chess engine. Chess engine reads the input file and writes its output on the output file which is read and sent to the robotic arm controller.

**4.3 ROBOTIC MANIPULATOR STRUCTURE:**

Industrial robotic manipulators could also be the option to accomplish chess playing task. But mostly robotic manipulators are designed with respect to specific automation purposes and for heavy material handling. Their motion trajectory and serving capability is restricted by their design. The chess playing robot is playing in humanly environment, it requires to be sophisticated and lightweight structure and human arm like motion while playing chess. While picking and placing chess piece, it should not collide with neighboring pieces. The sliding robotic arm was designed to avoid programming complexity of yaw motion. The intended column is accessed by the sliding movement of the robot and intended row is accessed by robotic arm joints movement. The robotic arm is sliding on an aluminum rail driven by two stepper motors. The aluminum was used to keep the weight of robot minimum. The four stepper motors were used as actuator of robot, while a DC motor was used for the opening and closing of the robot end effectors to decrease the complexity of the electronic circuit and program.The mechanical structure of robotic arm is as follows:

***Upper arm*** is connected with the base of the robot through shoulder joint. The other end is connected to the elbow (arm between shoulder and elbow). Shoulder joint is responsible for moving the whole upper arm. Shoulder’s actuator is connected with the upper arm via timing belt.

***Fore arm*** connects elbow and wrist just like a human arm. The elbow motor is responsible for the fore arm’s motion. The length of fore arm is 30cm.

***Wrist*** plays crucial role in lifting the piece from desired box without disturbing the neighboring pieces. When robotic arm reaches the desired box, it's upper and forearms reach above that box and then wrist moves downward for the accurate piece picking and placing.

***End effectors*** were specifically designed in such a way that its movement does not disturb the neighboring pieces. Its structure is made from aluminum and rubber inside to hold the piece firmly. The motion of gripper is controlled by DC motor. A screw is used to control the opening and closing of gripper jaw. When wrist reaches at specific height, the screw opens the gripper jaw to pick the piece and closes when it holds the piece.

4.4 CONTROLLER AND WORKING:

The software is divided into application specific and robotic-arm part. All the basic robotic arm related functionalities especially actuators controller speed, power supply and communication part control by this part. The 8-bit PIC16F877A microcontroller was used as primary controller of the robot. It is based on modified Harvard architecture. It is a 40 pin powerful microcontroller with 14kB flash memory, 356 Kbyte RAM. The low cost and powerful PIC architecture made it the best choice for this kind of application. The hardware also includes four stepper motor drivers on the same PCB board. The control unit is connected with computer through serial bus. The chess engine proposes the next move for robot. Octave is used for serial communication with robot controller. Chess engine writes its output on a text file. Octave reads the text file and sends it to controller of the robot over serial bus operating at 9600 baud rate. The control program parses the move which is sent in algebraic notation and calculates the number of cycles for each stepper motor to execute this move mechanically. There are two possibilities, either capturing a piece or simple placing a piece from one place to the other. In capturing case, robot first lifts the captured piece and places it out of chessboard at prescribed position and then placed the capturing piece on new box. In case of simple move, it transports the piece from one box to the other box. A small camera mounted on wrist section of the robotic arm which helps to recognize height and width of the chess pieces. When a box number is given to controller software, it finds box's column and row number and then starts its journey towards the box. The symmetrical design of chessboard simplifies the process of finding steps required for each actuator to reach any box. To perform any move, robot first recalls its current position, then it calculates each actuators movement required to reach the desired box. It slides over the relay and reaches calculated column, leans towards the calculated row and lift the piece. It repeats the process again, finds the destination box where it has to place the piece in a box or out of chessboard. Again it calculates the destination column and row and places the piece there. The last position of the robot is saved which will be used next time at the start of move implementation.

**CHAPTER 5**

**SYSTEM DESIGN**

The autonomous chess playing robotic system follows process to accomplish the desire goal.

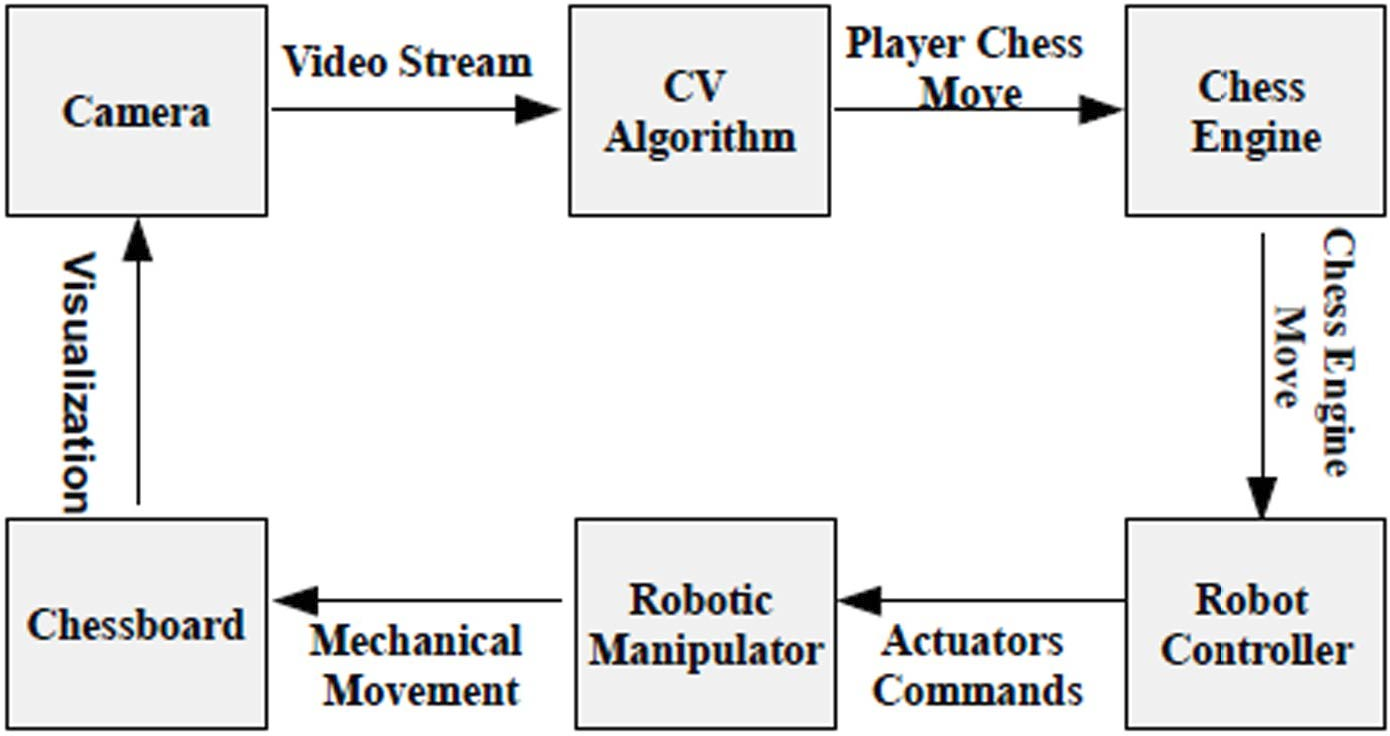
****

Fig 5.1 Process Flow

This is the camera setup and game setup that illustrates how this system going to work.

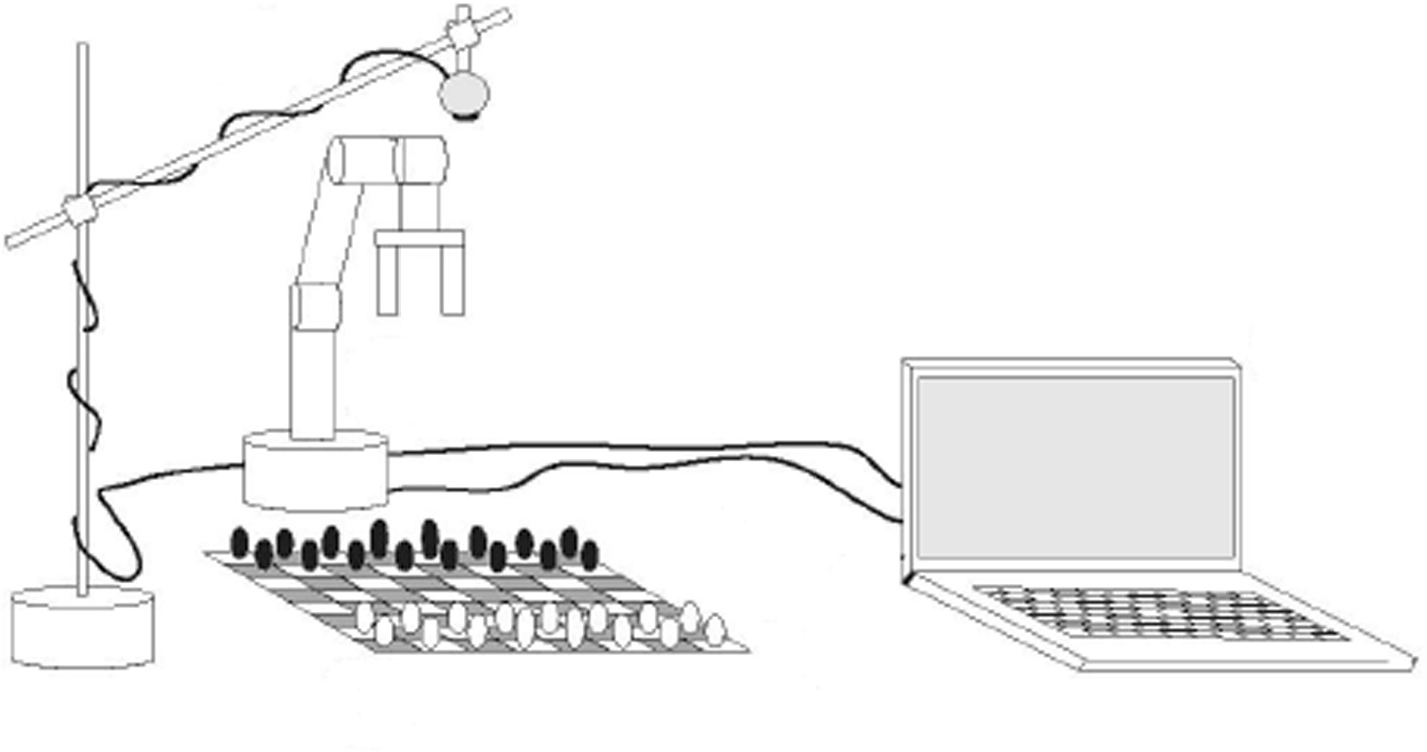


Fig. 5.2 Camera and Game Setup

This figure shows all of the segments of chessboard which are divided into 65 segments and converted into grayscale values that give maximum contrast.

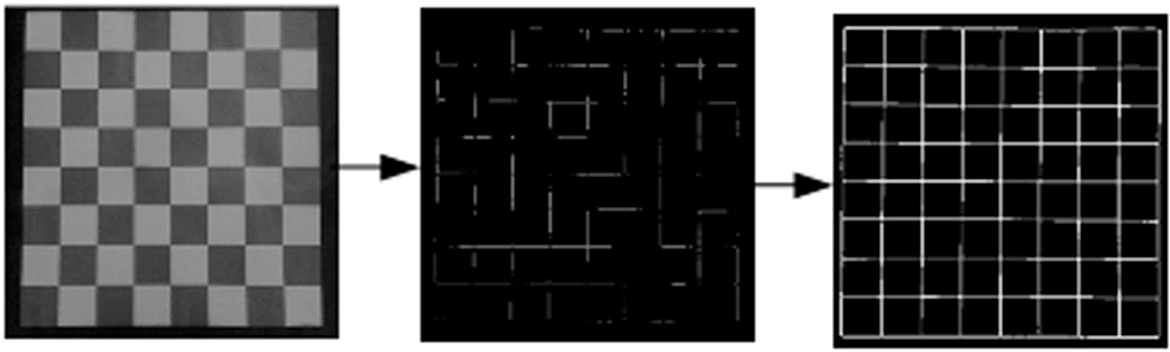


Fig. 5.3 Chessboard Detection

In move detection it captures the image before any hand motion detection save that image. After move detection it keeps in alter state and check every frame until hand motion vanishes. After that it subtracts the frame with previously saved image and shows the results.

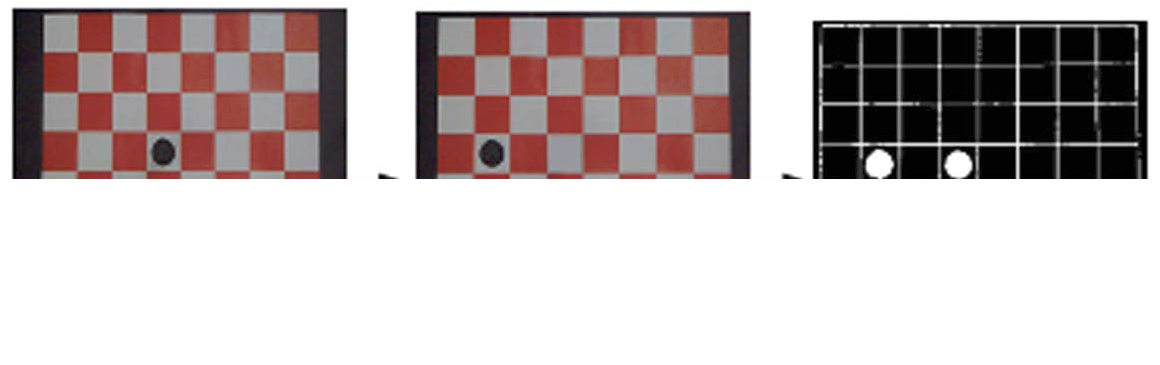


Fig. 5.4 Move Detection

This figure illustrates the algebraic notation and process control flow of the autonomous chess playing robotic system.

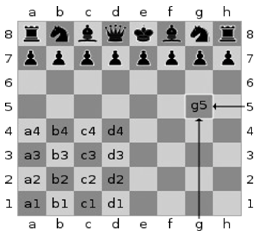
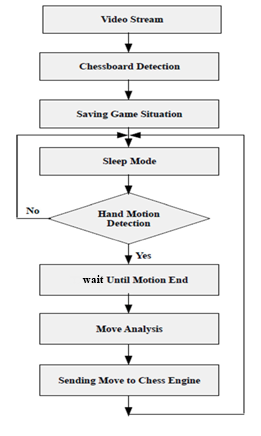
 

Fig. 5.5 Algebraic Chess Notation and Process Control Flow

**CHAPTER 6**

**ADVANTAGES & DISADVANTAGES**

*ADVANTAGES:*

1. **This system does not require human assistance.**
2. **This system is light weighted so the portability of it very high.**
3. **This system can be use for teaching purpose.**

***DISADVANTAGES:***

1. **The system cannot execute complicated moves such as The Castling and En-Passant.**
2. **This system cannot play against illegal move until it is corrected.**
3. Regular maintenance is required.

**CHAPTER 7**

**APPLICATIONS**

1. To Challenge Mental Capacity – Every human has different mental capability. As per human nature, human learn from the new things. This system can be used for challenge mental capability of human which helps to develop new skills.
2. **Teaching/Educational purpose** – The system can teach new players.
3. **Used as an Arbiter – This system can be used as a referee in game.**

**CHAPTER 8**

**CONCLUSION**

An autonomous robotic system described in this seminar is capable of autonomously playing chess against human or another robotic system without human assistance. The game situation analysis is entirely based on computer vision part. This topic gives an overview of the design and implementation of an autonomous chess playing robot, robotic system which is built from locally available, simple and low cost material. The software part is constructed in a separate module so that it can be easily changed or upgraded. This system also intend to include voice recognition system to provide user moves so that disabled persons can also be benefited from it in future.

**REFERENCES**

# Hafiz Muhammad Luqman, Mubeen Zaffar .“Chess Brain and Autonomous Chess Playing Robotic System” in 2016.

1. Cynthia Matuszek, Brian Mayton, Roberto Aimi, Marc Peter Deisenroth, Liefeng Bo, Robert Chu, Mike Kung, Louis LeGrand, Joshua R. Smith, Dieter Fox, “Gambit: An Autonomous Chess-Playing Robotic System” in 2011.
2. Emir Sokic, Melita Ahic-Djokic, “Simple Computer Vision System for Chess Playing Robot Manipulator as a Project-based Learning Example” in 2008.
3. Dennis Aprilla Christie, Tubagus Maulana Kusuma, and Purnawarman Musa, “Chess Piece Movement Detection and Tracking, A Vision System Framework for Autonomous Chess Playing Robot” in 2017
4. Bernd Jähne , "Image Formation", in Digital Image Processing, Heidelberg: Springer, 2005.
5. Shao-Yi Chien, Shyh-Yih Ma ; Liang-Gee Chen , "Efficient moving object segmentation algorithm using background registration technique," IEEE Trans. on Circuits and Systemsfor Video Technology, 2002.
6. T. Standage, The Turk: The Life and Times of the Famous Eighteenth-Century Chess-playing Machine. Walker, 2002.
7. M. Piskorec, N. Antulov-Fantulin, J. Curic, O. Dragoljevic, V. Ivanac, L. Karlovic , "Computer vision system for the chess game reconstruction", in Proc. of the 34th International Convention *(MIPRO)*, Opatija, 2001.

**BIBLIOGRAPHY**

1. FIDE, "Algebraic Chess Notation", Feb 12. 2016, [Online]. Available: https://en.wikipedia.org/wiki/Algebraic\_notation\_(chess)