Lego Plays Chess: A Low-Cost, Low-Complexity Approach to Intelligent Robotics

Michael Lanighan and **Jerod Sikorskyj** and **Debra Burhans** and **Robert Selkowitz**
Departments of Computer Science and Physics
Canisius College
Buffalo, NY 14208
{lanigham, sikorskj, burhansd, selkowir}@canisius.edu

Abstract

The design and implementation of a robotic chess agent is described. Shallow Blue, a competitor in the AAAI 2011 Small Scale Manipulation Challenge, is constructed with low-cost components including Lego NXT bricks and is programmed using Java and Lejos.

Introduction

The AAAI 2011 Small Scale Manipulation Challenge pits robots against each other in a game of chess (chiara-robot.org/Challenge). In the first chess challenge, which occurred at AAAI 2010, the field of competitors was dominated by fixed arm robotic platforms such as Gambit (Matuszek et al. 2011). However, these sophisticated platforms are often out of reach for institutions without a large robotics budget.

Shallow Blue (SB), named in honor of Deep Blue and for the minimal budget it took to construct, differs greatly from past entries. It is not an arm, but rather resembles a copier in the way it scans the board from above with a rolling bridge. In addition, it is built using off the shelf Lego Mindstorms components (mindstorms.lego.com).

Hardware

SB consists of five major components: a Lego mechanical structure, two associated NXT bricks (SB1 and SB2), computer vision mounts, and a control laptop. The mechanical structure consists of a bridge that rolls on top of a set of rails. Moving on top of the bridge is a wheeled tower affixed with a claw. This allows SB to have a combined three degrees of freedom (DOF), each DOF independent of the others. The distance between rails is the width of the board plus a five-inch neutral play area on either side. The rails and bridge feature yellow pegs that respectively demarcate row and column boundaries. The entire mechanical system is utilized by the two NXT bricks.

The first brick, SB1, is attached to the bridge structure. Connected to a pair of color sensors and motors, SB1 controls row movement and sensing on the left and right sides of the board. Each NXT motor contains an internal rotation sensor, capable of +/-2 degrees of accuracy. The second brick, SB2, is mounted on the tower. It is connected to three NXT motors. The first motor controls claw manipulation, while the second motor raises and lowers the claw. Finally, the third motor moves the tower across the board columns. Like SB1, internal rotation sensors are included for each motor. Additionally, one color sensor is used to sense column position relative to the bridge. Outside of the rails, our vision hardware scans the board.

The vision hardware consists of a Nikon CoolPix S570 digital camera mounted in a ring stand. The camera is elevated approximately two feet above the board in order to capture the full game state. A USB cable connects the camera to the control laptop.

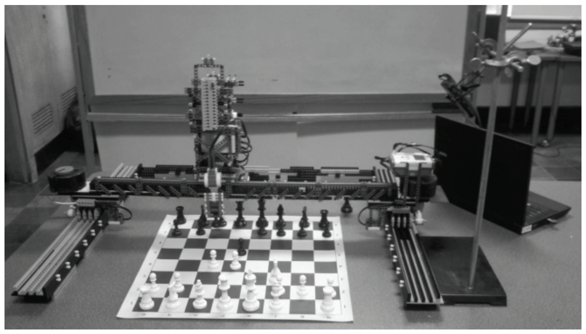
Figure 1: *Shallow Blue acquiring a pawn*

Copyright © 2011, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

The control laptop, a Dell Latitude E4500, sits outside the mechanical structure. Utilizing an Abe UB22S Bluetooth dongle, the laptop wirelessly communicates with the NXT Bricks. This computer centrally connects the hardware and software of the system.

Software

The software system is divided into four major subsystems: the Master Control Program (MCP), movement subsystems SB1 and SB2, chess reasoning, and vision. SB utilizes Java 1.6.0_21 for the MCP. SB1 and SB2 use LeJOS 0.85a, a firmware replacement for Lego Mindstorm NXT robots that provides a Java-based programming environment (lejos.sourceforge.net).

The chess reasoning subsystem utilizes a Universal Chess Interface (UCI) engine binary coupled with Java-based control software. Prototypes of the vision software subsystem use OpenCV (opencv.willowgarage.com) and ImageJ (rsbweb.nih.gov/ij), open-source computer vision libraries.

The computationally intensive subsystems of SB, such as the MCP, chess reasoning, and vision are run on the control laptop. By coordinating these separate subsystems, the MCP enables enhanced computing power and flexibility not available on an NXT brick. This centralized design allows for the optional use of additional external computing resources, such as a cluster. Additionally, the MCP acts as a host to the brick-based programs, SB1 and SB2, communicating which piece to move.

SB1 and SB2 each contain a listener NXT program and corresponding motor control logic to move the mechanical structure. The SB1 subsystem controls row movement while the SB2 subsystem controls both column and claw movement. In each case, the MCP wirelessly coordinates these movement subsystems for piece acquisition through Bluetooth.

Piece acquisition, however, needs to be informed by the chess engine before it can be initiated. Competition rules necessitate that choosing and physically completing each move must be done within two minutes. Utilizing existing chess software is prudent given the fact that designing a fast, grandmaster level computer chess program is not the goal of the competition.

Contemporary computer chess programs have evolved to have two major components: a chess engine where calculating the next best move is completed and a chess user interface that allows for player input/output, detailed game-state storage, and display. Communication protocols have been invented to talk between these two components (www.horizonchess.com). The three dominant protocols are: ChessBase, WinBoard/Xboard, and UCI (www.shredderchess.com). ChessBase is a closed protocol system while WinBoard/Xboard and UCI are both open.

The world's top five chess engines all utilize UCI, while only some of those five use the other protocols. We selected two free engines, Houdini 1.5a (http://www.cruxis.com/chess/houdini.htm) (closed source) and Stockfish 2.0.1 (www.stockfishchess.com) (open source), respectively ranked #2 and #3 worldwide for time controls of forty moves in four minutes (computerchess.org.uk). These chess engines, both advanced alpha-beta searchers, run on the control laptop. Custom-designed control software will parse engine output, store game-state details, and send the next move to our movement subsystem. This chess decision subsystem allows for maximum flexibility to use the best and fastest available chess engines right up to competition deployment. To predict a legal move, the chess decision subsystem will need an accurate vision subsystem for board state recognition.

Preliminary work on vision includes parsing camera images using ImageJ and OpenCV. Both can be called from the MCP using Java. The major challenge is detection of the initial board state, which can be mid-game. Move detection involves change between sequential images (for example, as described in the paper by Sokic and Ahic-Djokic 2008). Initial results suggest that the addition of NXT-connected color sensors and IR distance sensors mounted underneath the bridge may aid in piece recognition.

Conclusion

Although not infinitely applicable outside of chess, SB would easily transition to other styles of board games. The system could be applicable in industrial overhead systems, albeit with different components. Most importantly though, is that SB is proof that anyone, in theory, can build an intelligent robotic system on a minimal budget.

References

Matuszek, C., Mayton, B., Aimi, R., Deisenroth, M.P., Bo, L., Chu, R., Kung, M., LeGrand, L., Smith, J.R., Fox, D. 2011. Gambit: A Robust Chess-Playing Robotic System. To appear in *Proceedings of ICRA 2011*. Shanghai, China: IEEE.

Sokic, E. and Ahic-Djokic, M. 2008. Simple Computer Vision System for Chess Playing Robot Manipulator as a Project-based Learning Example. In *Signal Processing and Information Technology, 2008. ISSPIT 2008*, 75-79. Sarajevo, Bosnia and Herzegovina: IEEE.

Acknowledgments

Special thanks to the Canisius College CSC360: Intelligent Systems class (Spring 2011), as well as the Physics and Computer Science departments, especially Matthew Hertz.