Visual Chess Recognition
EE368, Spring 2015

Cheryl Danner, Mai Kafafy
cdanner@stanford.edu,mkafafy@stanford.edu

April 29, 2015

1 Project Goal

The goal of the proposed project is to correctly detect and identify a chessboard and the configuration of its pieces through the application of image processing techniques. Such an algorithm could be used to automatically record a game between two players without the need for a digital chess set, which can cost hundreds of dollars. In addition, image-based detection of chess pieces is a vital step in building chess-playing robots, as the playing strategy of the robot depends on its knowing the locations of its own chess pieces and the pieces of its opponent. Chess-playing robots can be used for fun and have furthermore been considered as an interactive toy that helps in developing the learning abilities of children .

2 Work Flow

At a high level, the algorithm will perform the following steps: chessboard detection and segmentation, determination of square occupancy, and recognition of the chess pieces. The subsequent list discuss these steps in further detail and are followed by a discussion of elements that could make the problem easier or more difficult. Implementation will be performed in MATLAB and OpenCV using still images from a digital camera.

1 The first step will be to detect the square pattern of the chessboard and identify the individual squares. Several studies have been conducted on this subject not only for the purpose of chess, but also for camera calibration using a chessboard pattern. The detection techniques can generally be categorized as corner detection and edge detection. The camera calibration process in [1] uses Harris corner detection and Canny edge detection with the Hough transform to identify a chessboard pattern from various angles. [2] presents an alternative algorithm for grid detection for camera angles between 30-90 degrees, and compares the method to the previously mentioned corner and edge detection methods.

2 After the individual squares have been identified, each square must be evaluated to determine if it is occupied or not, and, for squares that are occupied, the color of the piece that is present. This may be performed using a comparison to a reference image of an empty board in conjunction with gray level statistics for the potential occupancy region of each square.Paper [3] uses the point cloud above the chess board surface to determine the occupancy of the board. An anticipated challenge in this step is detecting a piece that has the same color as the background square, as noted in [4], which uses sums and differences of average gray levels, and [5], which modified the colors of the board to improve contrast, and [3] uses chess pieces with different colors.

3 Finally, feature recognition can be used to classify the chess pieces into one of six types: pawn, knight, bishop, rook, queen, or king. [4] and [5] detect the player moves by comparing the captured image of the chess board after the move with the stored image before the move. They assume that the locations of the chess pieces at the start of the game are entered to the robot manually. In our term project we will assume no prior knowledge of the locations of chess pieces, and the robot will have to detect it by itself from the captured image. Training images will be captured from different angles for each of the six types of pieces, with an anticipated fewer number of images needed for pieces with azimuthal symmetry.

3 Extra Challenges

Many complicating factors are likely to be encountered in the use of an image-based chess recognition product: viewing the board at an extreme angle, non-uniform lighting conditions, damage to the board or pieces, and extraneous objects in the scene. Some other inherent challenges are edges or corners being occluded by pieces and the standard use of the same two colors for squares and pieces. The objective of building in robustness to some of these conditions is viewed as a stretch goal. On the other hand, if some aspects prove to be overly complicated for the scale of this class, these factors can be limited or adjusted to make the problem tractable.

References

[1] A. De la Escalera and J. M. Armingol, "Automatic chessboard detection for intrinsic and extrinsic camera parameter calibration," *Sensors*, vol. 10, no. 3, pp. 2027–2044, 2010.

[2] K. Y. Tam, J. A. Lay, and D. Levy, "Automatic grid segmentation of populated chessboard taken at a lower angle view," in *Digital Image Computing: Techniques and Applications (DICTA), 2008.* IEEE, 2008, pp. 294–299.

[3] C. Matuszek, B. Mayton, R. Aimi, M. P. Deisenroth, L. Bo, and D. Fox, "Gambit: A robust chess-playing robotic system," in *Proc. of the IEEE International Conference on Robotics and Automation*, 2011.

[4] T. COUR, R. LAURANSON, and M. VACHETTE, "Autonomous chess-playing robot," *Ecole Polytechnique, July*, 2002.

[5] N. Banerjee, D. Saha, A. Singh, and G. Sanyal, "A simple autonomous robotic manipulator for playing chess against any opponent in real time."