CHESS STATE DETECTION

Bhavani B S
bhavani@stanford.edu

Chess is an amazing strategic game played by two players. Chess strategy consists of setting and achieving long-term positioning advantages during the game. One needs a lot of practice to master this. During the progress of a game one may not be sure as to which move would give better result. Using image processing techniques, we can detect the current state of the game and then apply an algorithm to suggest the next move.

I propose to develop an algorithm to detect the current state of a chess game which is oriented in 3 dimensional spaces. For this I need to capture pictures of chess board at different viewing angle and then process this to extract the required information [1]. I have come up with a strategy to tackle this. Have summarized below:

1. Segment the chess board: Remove the background details.
2. Detection of checkers: Use line detector to detect the boundary of chess board. Compute the edge equations.
3. Detect its orientation in 3d space: I will take advantage of the shape of the chess board to detect its orientation. In step 2 I will have the edge equation of the parallelogram. This parallelogram is obtained by 3D transformation of a square. I will compute this transformation and use it later in the algorithm.
4. Locate the center point of each square and label it: Using the above transformation I will detect the center of each square in the chess board for the given 3D orientation.
5. Determine the occupied squares: Have an 8x8 array and indicate occupied squares. Here overlapping pawns may cause problem. Need to tackle this.
6. Go to each of the occupied locations and identify the pawn present: use template matching considering orientation and size.
7. Once we know the current state of game we can apply an algorithm to predict the next best move

Main challenges:

1. 3D orientation of the pawns: Here I might have to scale and orient the templates accordingly and then use template matching [4]. Detection of asymmetric pawns.
2. Overlapping pawns: Identification of smaller pawns when they are partially or completely masked out by bigger ones [3].
3. When both chess board and pawns are of same colors: Identification of a pawn when viewed at a particular angle when some portion of the background is of same color as foreground [2].
4. Determining the edge of the chess board where the game begins : Labelling of squares when pawns are randomly positioned (will have to assume the front facing edge as the side where we start game)

I will implement this on Matlab.

Reference:

[1] Wang, V; Richard Green, "Chess move tracking using overhead RGB webcam", pp 299 – 304, 27-29 Nov. 2013

[2] Dhara, B.C. ; Chanda, B., "A Fast Interactive Image Segmentation to Locate Multiple Similar-colored Objects", Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIPG), 2011 Third National Conference, pp 25 – 28, 15-17 Dec. 2011

[3] Arteta, C. ; Lempitsky, V. ; Noble, J.A. ; Zisserman, A., "Learning to Detect Partially Overlapping Instances", Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on, pp 3230 – 3237, 23-28 June 2013

[4] Hinterstoisser, S. ; Lepetit, V. ; Ilic, S. ; Fua, P., "Dominant orientation templates for real-time detection of texture-less objects", Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference, pp 2257 – 2264, 13-18 June 2010