Journal Report 8
10/28/19-11/08/19
Kevin Fu
Computer Systems Research Lab
Period 1, White

Daily Log

Monday October 28

Attempted to iterate through orthophoto regions; ran into abnormally low canny edge pixel counts per square.

Tuesday October 29

Found error was because orthophoto from warpPerspective on Canny image sets the color range to 0-255, whereas the Canny image is binary (0 or 255). Added thresholds for minimum color to count as an edge pixel and minimum number of edge pixels per square to contain a piece (127 and 10, respectively).

Friday November 1

Changed minimum edge pixel threshold to be the average pixel value in the orthophoto rather than being exactly 127. (In other words, the strong half of the Canny pixels were counted.) Modified square labelling to only prompt on images deemed likely to have a piece by this method.

Wednesday November 6

Attempted using OpenCV's solvePNPRansac and projectPoints method to get a piece height dependent on the images' projection angle, rather than a hardcoded 150 pixels. Did not succeed.

Thursday November 7

Found error was in ordering of the points passed to solvePNP, fixed. Created graphical representation of the new scaling heights (see Reflection). Modified square labelling to use scaling heights. Labelled the pieces of the two test images I'd been working with this week.

Timeline

Date	Goal	Met
Oct 21	Gather and label images from chess videos with board-seg script	Modified goal to use orthophoto
Oct 28	Use perspective transform on board segmentation to identify pieces	Done
Nov 4	Gather and label images with orthophoto guesses considered	Labelled two chessboards for 70 images so far
Nov 11	Label roughly 30 more chessboards for 1000 training photos, and research and implement image augmentation	Not started
Nov 18	Construct a CNN to recognize augmented chess piece images	Not started

Reflection

Over the last two weeks, I strayed even further off the path to a piece-labelling CNN, but I think the time spent will be worth it. For one, the system can now judge for itself whether or not a piece is on a square: this will improve the accuracy of the final PGN writer and the rigor of the empty square test cases the CNN is trained on. Also, rather than labelling all 64 squares of every chessboard photo I find, I now only need to label a portion of them. For example, the high-angle case (shown below) only marked 14 squares as potentially having pieces and needing a label, of which 8 actually contained pieces. (The low-angle case was 9 for 56, significantly worse, but still less than the full 64.) Full projected heights shown in Figure 1.

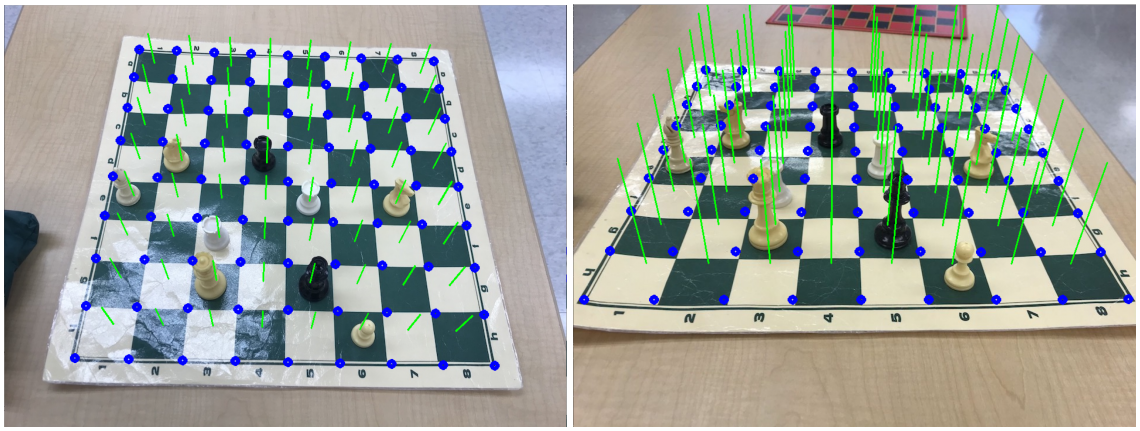

Figure 1: Projected max piece heights in green on high-angled chessboard (left) and low-angled chessboard (right).

There's a minor problem where on lower-angled images, even with the projected piece-heights, pieces which are not on the labelled square are still fully visible out of necessity. The two images in Figure 2 below illustrate this problem. The queen on the left is fully visible, but the empty square on the right has both a rook and queen partially visible in the frame. I may reduce my height estimate to cut off the top of taller pieces, or scale the projected height relative to how warped the orthophoto is.

Figure 2: A flaw with projected piece heights on low-angled images (two left). Comparison high-angle square on right.

In the coming weeks, I will address the low-angled image issue, and push forward on labelling many chessboards and implementing image augmentation. According to this timeline, I'll have data and a framework for a piece-recognition CNN by Thanksgiving break. I was originally planning to create my own architecture, but I'll look into pretrained CNNs like ResNet50 as well.