AAR: Project Proposal

Casper V. Kristensen - 201509411
Alexander Munch-Hansen - 201505956

26th March 2019

1 Envisioned Use Case

We wish to create an augmented reality multiplayer chess experience. Each player will be playing separately at their own chess board using their own physical pieces, while the opponent's pieces – which are not present physically – will be rendered through a see-through head mounted display like the Hololense. We envision the final product to look similar to this[1] project by Günther et al.[2].

2 Actual Plan

Instead of using a HoloLens, we can implement the system using a webcam and setup similar to that of the Augmented Reality course. We will utilise machine learning to recognise the board state from the camera image, which, most notably, involves identifying the different board pieces and their position on the board.

To begin with, getting the system working from the perspective of a single user does not require being able to separate black and white pieces, since each player will only play using their own physical ones. Once a method for detecting a single colour has been developed, however, expanding the dataset to include the other one should be trivial.

In addition to reducing the scope by colour, we will also start by implementing an algorithm for identifying only two or three different and easily distinguishable pieces like the pawn, knight, and king. Expanding the algorithm to the remaining pieces may prove difficult, since, for example, the original[2] pawn and bishop may look similar, in which case we will have to look into creating our own.

Another issue is occluding the virtual pieces with the real ones, allowing for a seamless augmented reality experience. To utilise our knowledge from

[1] `https://www.youtube.com/embed/Geyr95Nl8mc?start=4&end=23&autoplay=1`
[2] https://en.wikipedia.org/wiki/Staunton_chess_set

the previous course we would hope to use the Unity game engine for this task, however, as some OpenCV functionality is unavailable through the OpenCV-ForUnity module, we may have to use other frameworks for the object detection, and therefore also for rendering virtual objects, considering integration and operability.

Lastly, as shown in the video by Günther et al., showing where each piece is allowed to move upon picking it up or pointing to it could increase the usefulness of the system, as well as the scope of the project, if there is time to do so.

2.1 Milestones

1. Persuade lecturer or TA into letting us use cool new HoloLens rather than boring old webcam.
2. Implement board detection and homography/warping.
3. Gather initial dataset for 2-3 pieces of a single colour based on previous step and simple script.
4. Implement and train machine learning algorithm for classification of pieces in reduced dataset.
5. Virtual object rendering and occlusion between real and augmented pieces.
6. Expand dataset to include both black and white pieces.
7. Integration with open source chess engine to show player optimal move.

References

[1] Maciej A. Czyzewski. An extremely efficient chess-board detection for non-trivial photos. *CoRR*, abs/1708.03898, 2017.

[2] Sebastian Günther, Florian Müller, Martin Schmitz, Jan Riemann, Niloofar Dezfuli, Markus Funk, Dominik Schön, and Max Mühlhäuser. Checkmate: Exploring a tangible augmented reality interface for remote interaction. In *Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI EA '18, pages LBW570:1–LBW570:6, New York, NY, USA, 2018. ACM.

[3] Frédéric Rayar, David Boas, and Rémi Patrizio. Art-chess: A tangible augmented reality chess on tabletop. In *Proceedings of the 2015 International Conference on Interactive Tabletops & Surfaces*, pages 229–233. ACM, 2015.