

CSCI508 Advanced Perception and Computer Vision Final Project Proposal - Augmented Reality N Queens Problem

Joseph Greshik

February 22, 2020

Project Proposal

For my final project I would like to develop a python application, 8NQ, that provides real-time feedback to the progress of a user solving the n queens problem on a real-life chess board.

The program will take in video feed of a chess board that the user is solving the puzzle on. The program will process the chess board information and generate visual cues within the video feed of the game to help the user correctly solve the puzzle. When the board contains a solution for the puzzle the video feed will have a message displayed to congratulate the user.

If time permits, I would like to have 8NQ also project solutions onto the board given an initial board layout.

N-Queens Puzzle

The n queens puzzle is the problem of placing n queens on an $n \times n$ chessboard where no two queens are threatening each other. The problem has solutions for all chess boards where $n \neq 2$ and $n \neq 3$. 8NQ will target the 8 queens puzzle, which uses a normal 8×8 chess board fully. An example solution to the 8 queens puzzle is shown in figure 1.

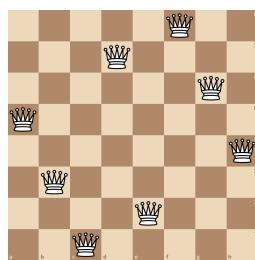


Figure 1: One of 92 solutions to the 8 queens puzzle.

Problem Approach

The deliverables required to finish this project will consist of the following:

1. Chess board detection by square (Feb 29)
2. Chess board square queen detection (March 21)
3. N queens algorithm to validate / invalidate placements (March 28)
4. Visual cues outputted to video feed (April 4)

Chess board detection by square

There are many existing frameworks for processing checker boards using OpenCV in python. I plan to use an existing method for partitioning the chess board into 64, normalized squares for piece recognition.

I plan to use my own chess board and pieces for the configuration of 8NQ. Instead of queen chess pieces I will use pawn pieces since they come in sets of 8.

All processing of the board will be performed on an orthorectified version of the board scaled to 512 pixels, resulting in individual square images of size 64×64 pixels for piece recognition.

Chess board square queen detection

In order to process chess squares for piece detection I plan to train a convolutional neural network which performs the binary classification of pawn present / pawn not present for a 64×64 image.

I will train the network on panchromatic orthorectified images of board squares. The input training data will consist of several different pawn piece types, lighting environments and board perspectives.

The use of this robust neural network should allow for the use of 8NQ in a variety of lighting environments and camera angles.

Error thresholds will need to be set as the project develops.

N queens algorithm to validate / invalidate placements

I plan to populate a simulated board data structure from the information gathered from board processing and check it for validity using one of many existing n queens problem algorithmic approaches.

The validity check will flag each piece on the chess board as either valid or invalid. 8NQ will output visual cues according to each detected piece's validity flag.

The first solution for checking placement validity I will try to integrate into 8NQ is backtracking. If backtracking is too slow to allow at least 15 fps for the processing of the output video feed then I will look into other more efficient solutions to the n queens validity check.

Visual cues outputted to video feed

Visual cues I would like to implement would be simply projecting semi-transparent cubes over board pieces, base color depending on the validity of each piece.

This will require mapping the simulated board within 8NQ to the real chess board, which should be pretty straight forward after board detection by square is finished.

Motivation

I think this will be a very fun project to work on since my first self-defined computer science project at Mines was solving the n queens problem generally in a simulation. Seeing it being done in real time using augmented reality will be very cool.