Augmented Reality Chess

3D Photography Project Proposal
Supervised by: Olivier Saurer
March 6, 2015

GROUP MEMBERS

Radek Danecek Alexander Lelidis

I. DESCRIPTION OF THE PROJECT

The goal of this project is to create a augmented reality chess game. We will use two cameras - one RGB-D (Microsoft Kinect) camera and one thermal camera. The Kinect will be used to track the paper checkerboard of the game which will be lying on a table while the thermal camera will be used for interactions with the players. (When a user touches the surface of the board, he makes it warmer for a short while). The depth video of Kinect will help to determine whether or not is the user's hand touching the checkerboard or not. The scene will be displayed on a screen augmented with the checkerboard and the figures along with all the game animations.

II. WORK PACKAGES AND TIMELINE

The game will run on PC. We will program in Python using wrappers for both OpenCV and OpenGL as well as our own C++ code. The important components of our project are: camera calibration, pose estimation, checkerboard tracking, user interaction detection, chess engine and game rendering. Following sections describe each component in detail. We expect that every component will be working by April 20th (the midterm presentation) to some extent. By May 18th (final presentation date) we would like to have a fully interactive augmented chess game.

A. Camera calibration

A paper checkerboard lies on a table and it will be used to calibrate the RGB camera of our Kinect. The thermal camera will be calibrated similarly [2]. The black squares on the checkerboard should be a bit warmer. If needed the checkerboard will be lit with a light increasing the temperature differences of the black and white squares. The calibration will be done using OpenCV [4]. After the initial calibration, the intrinsic and extrinsic parameters of the camera will be known.

Responsibility of: Alex Lelidis

B. Checkerboard tracking and pose estimation

The core idea of augmented reality chess is to have an online projection over a video in real time. This could be done using a device such as Google Glass or any other head-mounted camera/display device. Everything should look realistic as the user moves (and camera pose changes). In this project we will use a mere, possibly moving camera and display the output on a screen. In the beginning, we will assume that our camera (Kinect) is static and given enought time we might also try to extend it to moving camera. For this we need to keep tracking the checkerboard and updating the pose estimation. This consists of following steps:

1) feature extraction
2) feature matching
3) find epipolar geometry
4) estimate the camera pose

All of this will be done using OpenCV.

Responsibility of: Radek Danecek

C. *User interaction detection*

The thermal camera will be used to detect user input. By touching the surface of the checkerboard, the user makes it slightly warmer in that point and therefore detectable for the thermal camera. This is one of the greater challenges of the project as we need to properly distinguish the heat signatures of the touches on the checkerboard. We also need to filter out the occluding hand before and after the touch happens. We will use the depth input provided by our Kinect. The detection will have to be fast because we want our game to run in real time. Therefore we will use simple background subtraction followed by segmentation as in [3]. This part will be written in C++ using OpenCV and wrapped by Python interface.

Responsibility of: Radek Danecek & Alex Lelidis

D. *Chess engine*

We will use Sunfish - open source Python chess engine [1].

E. *Game rendering and animations*

All the rendering and animations will be done in OpenGL either using a Python wrapper using C++ OpenGL code.

Responsibility of: Radek Danecek & Alex Lelidis

F. *Timeline*

9. 3. - 15. 3.	Setting up the project, get familiar with the HW and SW
16. 3. - 22. 3.	Camera calibration
23. 3. - 29. 3.	Checkerboard tracking and pose estimation
30. 3. - 5. 4.	User interaction detection
6. 4. - 12. 4.	Game rendering, animations
13. 4. - 20. 4	Debugging, tweaking, preparation for presentation, mid-term presentation
18. 4. - 17. 5	More debugging, tweaking, improvements, possibility of extension for moving cameras.
18. 5.	Final presentation

III. Outcomes and Demonstration

The result of the project will be an interactive augmented reality chess game. The chess game will be playable in real time. Part of our final presentation will either be a gameplay video or online demonstration of the game.

References

[1] Thomas Ahle. Sunfish chess engine, 2015.

[2] Yuncong Chen and Will Warren. 3d fusion of infrared images with dense rgb reconstruction from multiple views - with application to fire-fighting robots. 2014.

[3] Daniel Kurz. Thermal touch: Thermography-enabled everywhere touch interfaces for mobile augmented reality applications. In *Mixed and Augmented Reality (ISMAR), 2014 IEEE International Symposium on*, 2014.

[4] Open Source Library. Opencv, 2015.