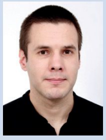
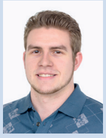
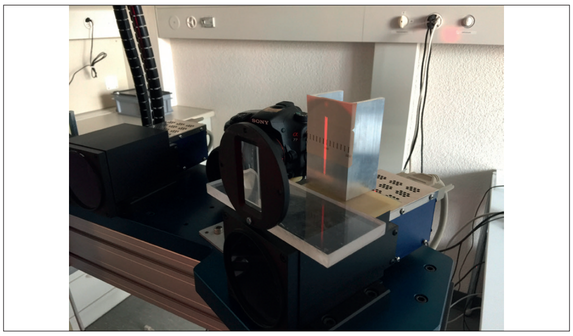



Roberto Cuervo Alvarez | Konrad Höpli

Graduate Candidates	Roberto Cuervo Alvarez, Konrad Höpli
Examiner	Prof. Oliver Augenstein
Co-Examiner	Reto Bättig, m&f Engineering, Fahrweid, ZH
Subject Area	Software
Project Partner	Clinic of Masticatory Disorders from the Center of Dental Medicine of the University of Zürich, Zürich, ZH

Jaw Viewer

Displaying jaw movement for medical diagnosis



3D Cameras

Jaw Viewer preview

Introduction: The Clinic of Masticatory Disorders offers treatment for facial pain and mandibular joint problems. For better diagnosis, the clinic has developed a proprietary 3D camera system (Optis) to record the patient's mastication movement. They are also able to extract bones from an MRI image of a patient and store them as Stereo Lithography (STL) files. In another proprietary software called TMJViewer, these resources can be merged to display the movement of the teeth and bones in a 3D animation, which is used for medical analysis. The current process to achieve this goal requires multiple systems, time consuming manual input, and is therefore error-prone. In addition, the current solution does not allow to display movement data in real time, which complicates the diagnosis process, because immediate feedback to the patient is not possible. The goal of this thesis is to develop a single, OpenGL-based application that simplifies the above process and further allows real time data analysis.

Approach/Technologies: To get familiar with OpenGL as well as various aspects of graphics rendering, we consumed some literature on the topic and walked through a tutorial that implemented Phong shading in OpenGL. We then used this code as a base for our project and extended its functionality according to requirements and thus obtained a working C++ prototype for our project. For that we imported anatomical objects (STL files) and transferred them to OpenGL as vertex buffer objects. In a second step, we were then using movement information from Optis to calculate rigid transformations which were afterwards applied to vertex data within a vertex shader. Finally, we refactored the prototype before integrating it into a new WindowsForms application providing a graphical user interface for its configuration. Due to an incomplete specification of the Optis calibration process, the displayed movement ended up not being completely accurate and a lot of reverse engineering was required to reach the resulting version of our software.

Result: This project resulted in an application consisting of a library component for the graphical display embedded into a minimalistic GUI for configuration. As the functionality of the existing application is not yet fully implemented, the software is not suited for productive use. However, it serves as a decent base for a replacement of the existing application and can be used – in combination with our documentation – as a solid introduction to OpenGL.