Technische Universität München
Fakultät für Informatik
Prof. Gudrun Klinker, Ph.D.
Andreas Langbein, Adnane Jadid, David Plecher

Summer Term 2018
Exercise Sheet 8
June 6, 2018

Introduction to Augmented Reality

Exercise 15 (P) Putting it all together

After we now have finished our marker tracking and rendering programs, it's time to put them together. Your task is to make the snowman move and rotate according to the marker movement. The easiest way to achieve this is to call a marker-tracker method in the rendering loop of GLFW.

Note: If you want to display OpenCV windows, you still have to call `cv::waitKey` at the end of the idle function, otherwise, you won't see them. *Note:* You can use `glLoadTransposeMatrixf()` to convert between column-major and row-major matrices or implement your own solution.

Exercise 16 (P) Background video

In an Augmented Reality (AR) application, real and virtual information are merged. We should therefore also render the video data, in addition to the virtual snowman. For this, images have to be passed from the tracker part to the renderer part of the program. The basic steps to integrate the video as background into your rendering window are:

- Set the pixel zoom and storage parameters (`glPixelStorei()`, `glPixelZoom()`).
- Store default projection matrix and disable the depth test (`glDisable()`).
- Set an orthogonal projection matrix covering the entire screen (`glOrtho()` with near = -1 and far = 1).
- Set the raster position (`glRasterPos2i()`).
- Call `glDrawPixels`, passing the pixel buffer from the `cv::Mat`.
- Restore the original projection matrix and re-enable the depth test.
- Load the modelview matrix of your marker and draw the snowman (as in Exercise 14).