**Advanced Topics in Computer Graphics II**

Universität Bonn
Institut für Informatik II
October 22, 2019

Winter term 2019
Prof. Dr. Reinhard Klein
Alexander Dieckmann, Lukas Bode

# Sheet R1 - Practical Rendering Project

## Project Introduction

During the first part of the lecture you are required to do a practical project in **groups of 2** (and a single group of 3 for an odd number of students). The project shall give you further insight into the process of material acquisition and rendering an SVBRDF in real-time. We will also compare two different methods for capturing an SVBRDF.

There will be scheduled meetings and you are expected to be able to present the results regarding the corresponding milestone. Thus, in every meeting at least one member of **each group has to be present** and capable of presenting. Each member of a group has to be able to explain the basics of what every single part of the group's code does. Therefore, we encourage you to talk about your code in the group. You have to **complete every single milestone in time** in order to be allowed to take the exam. The meeting dates and milestones are described below in more detail.

Your pipeline has to be written in **C++ using OpenGL**. We strongly recommend to use **CMake** as build system and **git** as version-control system. You are free to use libraries which provide basic functionality (GLEW, Glad, GLFW, GLM, NanoGUI, Assimp, tinyOBJ, ...), however, we expect you to implement the main parts of the pipeline by yourself.

You can find lots of tutorials about C++, OpenGL, and the mentioned libraries on the internet. Regarding C++ and OpenGL in general you can take a look at `https://learnopengl.com/`. It covers basically everything you need to know for this lecture and much more. More information about specific parts of C++ can of course be found at `https://en.cppreference.com/w/`. CMake and git have good tutorials on the respective websites: `https://cmake.org/cmake-tutorial/` and `https://git-scm.com/book/en/v2`.

If you have problems, questions, ... please write an email to the mailinglist (**Vl-atcg2@lists.iai.uni-bonn.de**) or the tutors (**lbode@cs.uni-bonn.de**, **dieckman@cs.uni-bonn.de**).

## Schedule (subject to change)

During the next 6 weeks, we will have scheduled meetings **16:00 - 17:30 o'clock in room 3.005**. The dates are:

- Tue, 22.10.19 Intro
- Tue, 05.11.19 Milestone 1
- Tue, 19.11.19 Milestone 2
- Tue, 26.11.19 Milestone 3
- Tue, 03.12.19 Milestone 4 & Showcase

## Milestones

The following milestones have to be completed in time.

**Milestone 1** As a first milestone, you have to capture a Ward SVBRDF using the Tac7 scanner. In the first meeting on 22.10., we will show you how to do it by exemplary scanning an arbitrary material. During the following two weeks you are expected to make an appointment with the tutors to scan a material on your own. The Tac7 outputs an .axf file containing the material information.

**Milestone 2** You have to be able to render the captured material using your own real-time renderer. The camera should be moving and there should be some mechanism (button press, ...) to pause the movement for further inspection. We will provide C++ code for parsing the .axf files.

**Milestone 3** Valentin Deschaintre et al. have published a deep learning approach to capture a Ward SVBRDF using a single flash smartphone image. This is an alternative approach to material acquisition using the Tac7 scanner. In order to compare both approaches, you have to read the paper, take a flash photo using your smartphone, run the neural network and render the resulting SVBRDF using your pipeline. The paper, source code, and trained model can be found at `https://team.inria.fr/graphdeco/projects/deep-materials/`.

**Milestone 4** Finally, you have to implement one extension for the pipeline you have implemented so far, that improves it in a meaningful way. In the following section, we have provided some exemplary ideas for extensions. If you have other ideas, just write an email or talk to us.

## Possible Extension Points

In addition to the fixed milestones 1, 2, and 3, **every group has to implement some extension** of the rendering pipeline. Examples of such extensions would be:

- Parallax and displacement mapping
- LEAN/LEADR mapping (or similar)
- Virtual point lights environment mapping
- Support of point set surfaces

Feel free to suggest other interesting extension ideas.

**Good luck !**