Prof. Dr. Leif Kobbelt
Dr. Jan Möbius
Patrick Schmidt, Moritz Ibing

Visual Computing Institute | RWTH AACHEN UNIVERSITY

Basic Techniques in Computer Graphics

Assignment 9

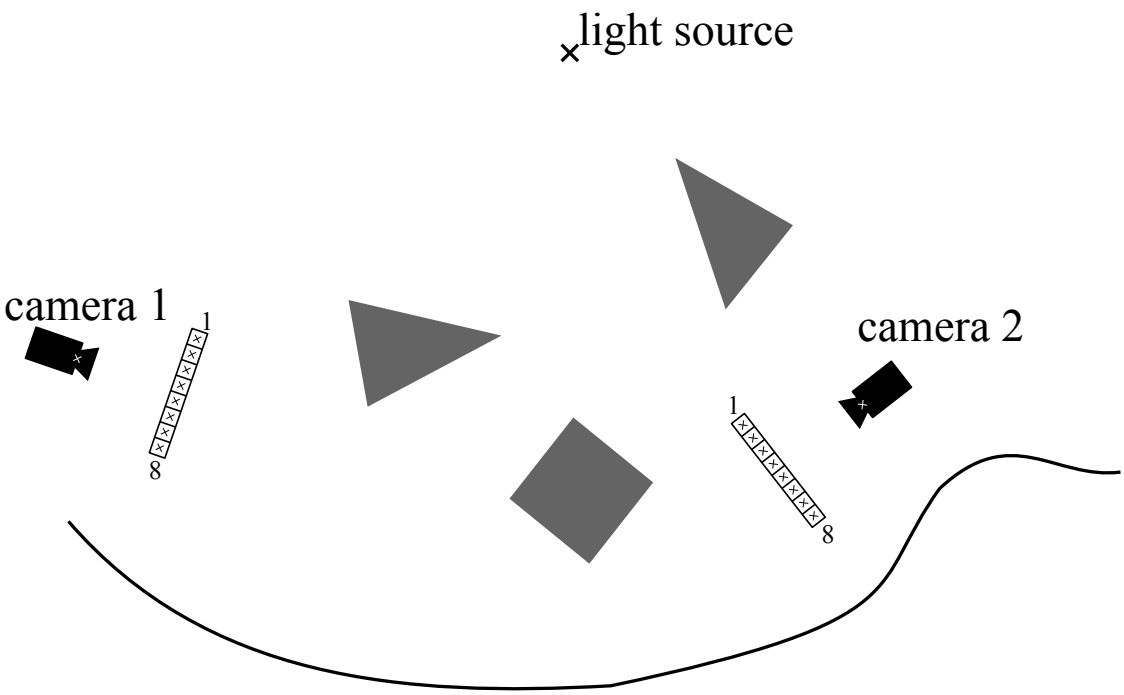
Date Published: December 10th 2019, Date Due: December 17th 2019

- All assignments (programming and text) have to be completed in teams of 3–4 students. Teams with fewer than 3 or more than 4 students will receive no points.
- Hand in **one solution per team per assignment**.
- Every team must work independently. Teams with identical solutions will receive no points.
- Solutions are due 14:30 on December 17th 2019. Late submissions will receive zero points. No exceptions!
- Instructions for **programming assignments**:
  - Download the solution template (a zip archive) through the Moodle course room.
  - Complete the solution.
  - Prepare a new zip archive containing your solution. It must contain exactly those files that you changed. **Only change those files you are explicitly asked to change in the task description.** The directory layout must be the same as in the archive you downloaded.
  - Upload your zip archive through Moodle before the deadline. Use the Moodle group submission feature. Only in the first week (when Moodle groups have not been created yet), list all members of your group in the file `assignmentXX/MEMBERS.txt`. Remember, only one submission per group.
  - Your solution must compile and run correctly **on our lab computers** using the exact same `Makefile` provided to you. Do not include additional libraries and do not change code outside of the specified sections. If it does not compile on our machines, you will receive no points.
- Instructions for **text assignments**:
  - Prepare your solution as a single pdf file per group. Submissions on paper will not be accepted.
  - If you write your solution by hand, write neatly! Anything we cannot decipher will receive zero points. No exceptions!
  - Add the names and student ID numbers of all team members to every pdf.
  - Unless explicitly asked otherwise, always justify your answer.
  - Be concise!
  - Submit your solution via Moodle, together with your coding submission.

Prof. Dr. Leif Kobbelt
Dr. Jan Möbius
Patrick Schmidt, Moritz Ibing

Visual Computing Institute | RWTH AACHEN UNIVERSITY

Exercise 1 Shadow Volumes [12 Points]



Consider the scene given above, containing a light source, three occluders (grey objects) and two cameras, where the center of projection is given by the white cross inside the camera.

(a) [3 Points]

Use a ruler to draw the shadow volumes generated by the light source and the occluders. Clearly mark which sides of the shadow volumes are *front-facing* towards camera 1 and which sides are *back-facing*.

(b) [3 Points]

Explain the relationship between ray casting and the Stencil Buffer-based shadow volume algorithms.

(c) [3 Points]

Use ray casting through the centers of the pixels (small black crosses) to compute for every pixel (1-8) the Stencil Buffer entry assigned to it by the normal shadow volume algorithm. Specify those entries for both cameras.

(d) [3 Points]

What do you notice looking at the Stencil Buffer entries for camera 2? Explain why this happens and how to fix the problem.

Prof. Dr. Leif Kobbelt
Dr. Jan Möbius
Patrick Schmidt, Moritz Ibing

Visual Computing Institute | RWTH AACHEN UNIVERSITY

Exercise 2 Volume Ray Casting [12 Points]

You are computing the perceived color of a viewing ray through a volume using Volume Ray Casting. Taking discrete samples along the ray, you find the following colors c_i (represented as monochromatic gray values) and opacities α_i (where 0 is fully transparent and 1 is fully opaque):

i	0	1	2	3	4	5
c_i	0.0	1.0	0.5	0.0	0.8	0.0
α_i	0.5	0.0	1.0	0.5	0.2	0.0

The index $i = 0$ represents the sample closest to the viewer and $i = 5$ the sample farthest away from the viewer.

(a) Back-to-Front [6 Points]

Compute the color along the ray using back-to-front compositing. Write down all computation steps.

(b) Front-to-Back [6 Points]

Compute the color along the ray using front-to-back compositing. Write down all computation steps. You may terminate early as soon as no further changes to the computed color value are possible.

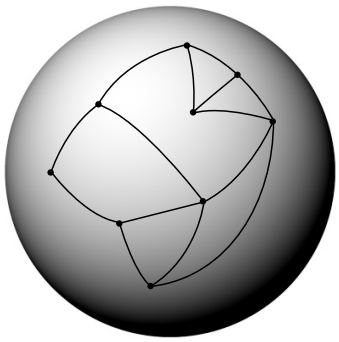
Prof. Dr. Leif Kobbelt
Dr. Jan Möbius
Patrick Schmidt, Moritz Ibing

Visual Computing Institute | RWTH AACHEN UNIVERSITY

Exercise 3 Meshes [16 Points]

(a) [3 Points]

Consider the depicted closed mesh (drawn on a spherical surface to underline its closedness). Depict the dual mesh of this mesh (in a way that every vertex, edge, and face is visible, i.e. don't draw on the backside of the sphere). Remember that duality in its core is a purely topological concept, i.e. you can place the dual vertices arbitrarily in the primal faces, edges don't have to be straight, faces don't have to be planar.



(b) [8 Points]

In the lecture you have seen how the average vertex valence of a triangle mesh as well as the relation between the number of faces and the number of vertices in a triangle mesh can be derived from the Euler formula. Remember that the Euler formula not only holds for triangle meshes, but also for arbitrary polygon meshes. Derive the average vertex valence of a closed hexagon mesh (of genus 1) as well as the relation between the number of faces and the number of vertices in such a mesh from the Euler formula.

(c) [5 Points]

Consider the depicted closed triangle mesh with 19968 edges. What is the genus of this object? Derive how many vertices and triangles this mesh has got.