

UNIVERSITÄT DES SAARLANDES
DR.-ING. HENDRIK P.A. LENSCH
MAX PLANCK INSTITUT INFORMATIK
ART TEVS (TEVS@MPI-INF.MPG.DE)
BORIS AJDIN (BAJDIN@MPI-INF.MPG.DE)
MATTHIAS HULLIN (HULLIN@MPI-INF.MPG.DE)



7. FEBRUARY 2008

COMPUTER GRAPHICS I ASSIGNMENT 11

Submission deadline for the exercises: Thursday, 14th February 2008

Rule: Solutions have to be submitted in the lecture room before the lecture.

11.1 Volume Rendering (50 Points)

- Describe informally what is Volume Rendering and which approaches exists to render a volume? (10 Points)
- In the previous exercises you have computed the normal of a triangle based on its vertices. Assume you can classify each voxel of your volume into transparent and opaque. You want to render the opaque part of the volume with applied phong shader. How you would compute the normals required for the lighting computation? (10 Points)
- Assume you have a volume with very low resolution. What would happen to your normals? What would be one of the ways to prevent this problem? (10 Points)
- The absorption law is given by:

$$\alpha = L_0 \cdot e^{-\mu \cdot d},$$

where L_0 is the initial radiance of the ray, d is the thickness of the medium and μ is absorption constant, also called absorbance. You want to render a volume with spatially varying attenuation (i.e. each voxel has a value σ_a which represents how much light is absorbed by this voxel). Derive now an equation which computes the absorption term α for spatially varying attenuation (you can assume that the rays are propagated linearly in the volume). (20 Points)

11.2 Splines (50 Points)

In this exercise you have to implement an interpolation with the help of a Cubic Spline for camera transformations in OpenGL. To start, download the slightly changed framework of the exercise sheet 9 from http://www.mpi-inf.mpg.de/departments/d4/teaching/ws2007-2008/cg/skeleton_splines.tgz.

- Make clear that you are familiar with Cubic Splines. You can find more information on <http://mathworld.wolfram.com/CubicSpline.html>.
- Implement the `Spline::create()` method which do prepare the parametrisation of the spline curve for further interpolation (35 Points).
- Implement the `Spline::operator()` operator for sampling the spline curve (15 Points).

If your implementation is correct you should be able to receive a smooth movement of the camera around the teapot.