Assignment 4: Pre-integrated 3D Volume Rendering and Time Step Selection TEMPORARY - WILL BE UPDATED SOON FOR USE WITH THE FRAMEWORK

Scientific Visualization 2022/23 (WMCS018-05.2022-2023.1A)
v1.0

October 4, 2022

1 General Information

The assignment is designed to be addressed alongside the lectures, with the individual tasks covering what has been discussed in the respective week. Such topics as Pre-integrated Volume Rendering, Time Step Selection, and Overlay Rendering are covered in this assignment. Update your private repository using instruction from the "Git guide".

2 Tasks

Task 1 – Pre-integrated Volume Rendering

The task consists of two parts: (1) the generation of a pre-integrated transfer function in a pre-computation step, and (2) the usage of this transfer function during raycasting. The idea here is to split the numerical integration into two integrations: one for the continuous scalar field and one for the transfer function.

Find the skeleton shader (`Pre-integratedVR.glsl`) in shaders. The skeleton shader `Pre-integratedVR.glsl` already contains a raycasting-based volume renderer. Your task is to fill in the code for compositing (your solution from the previous assignment can be reused) and to take care of the `TODO` comments. Find the file `pre-integration.cpp`. Your task is to fill in the code for pre-integration and to take care of the `TODO` comments.

For further information, please refer to the Eurographics tutorial notes *"L5-Real-Time-Volume-Rendering"* (pp 96–102) that you can find on Brightspace under "Additional Reading Material".

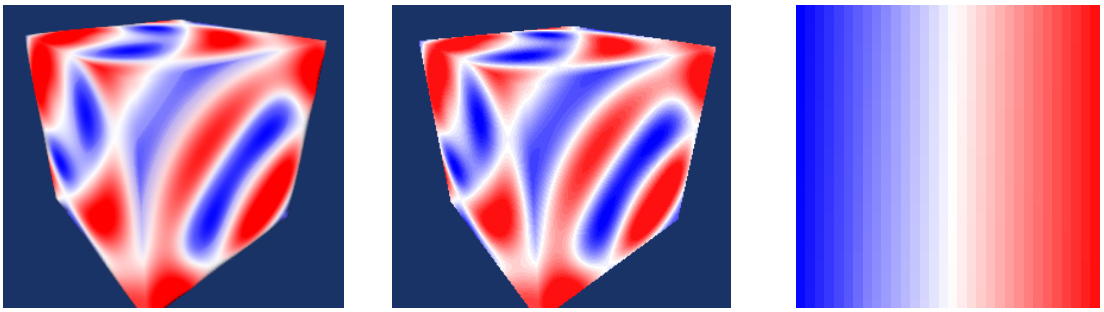
(a) **Pre-integrate Transfer Function (6 points)**
Modify the C++ program to pre-integrate the provided transfer function across all possible pairs of scalar values (s_f, s_b) and store it in a 2D RGBA color table. Output this pre-integrated (2D) transfer function in two different ways:

- A 2D color image (you may use the `writeImage` function for saving the image)
- A text string that can be directly used in GLSL (to be used in the shader for the next sub-task).

The color table with dimensions $32 * 32$ or $16 * 16$ is sufficient to demonstrate the concept, you don't have to use $256 * 256$ table.

Hint: for this you can use the compositing function you implemented for the previous task on volume visualization.

(b) **Use Pre-integrated Transfer Function (6 points)**
Modify the provided code to use the pre-integration table you generated in the previous sub-task. Generate screenshots with different step sizes and compare the differences between standard and pre-integrated rendering. Compare your results with those shown in Figure 1.



(a) Accumulation (b) Pre-integration (c) Lookup table

Figure 1: Accumulation vs. pre-integration using a lookup table.