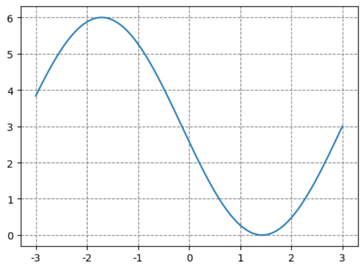
UNIVERSITÄT STUTTGART
Institut für Visualisierung und Interaktive Systeme (VIS)

Thomas Ertl, Steffen Frey *Stuttgart, 2019-06-21*

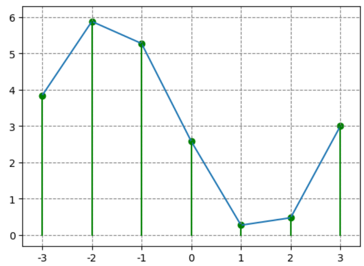
Scientific Visualization (Assignment 10)

Exercise 10. 1 [6 Points] Pre- and Post-Classification

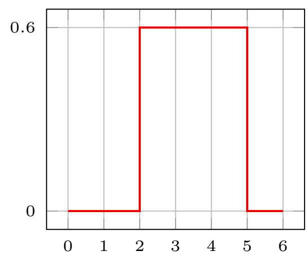
Transfer functions allow users to adjust visual feature classification of data during volume rendering. For direct volume rendering, we assume that discrete samples of the volume represent a continuous scalar field and thus we have to reconstruct the signal. For this task, let us assume a continuous 1D signal (Figure 1a) that was discretized on a uniform grid by shooting rays through the volume (Figure 1b). You are free to choose the resolution of the grid when doing interpolation, i.e., reconstructing the signal. Note that according to *Nyquist*, the sampling frequency for reconstruction has to be at least twice as high as the signal frequency.



(a) Continuous signal



(b) Discretized signal



(c) Transfer function

Your task is to apply both pre-classification and post-classification to the discrete signal using the transfer function $\mathrm{tf} : x \mapsto \alpha$ depicted in Figure 1c. *Pre-classification* refers to applying the transfer function to the discrete samples and then interpolating to reconstruct the signal. *Post-classification* means that the signal is first reconstructed through interpolation, and then the transfer function is applied to the reconstructed signal.

1. Draw a plot of the resulting α values (NOT the accumulated α) along the ray using the pre-classification scheme.

2. Draw a plot of the resulting α values along the ray using the post-classification scheme.

3. Which classification-scheme do you think is more appropriate and why? Think about what the perfect solution computed on the continuous signal would look like.

Exercise 10. 2 [4 Points] Composition Schemes in Direct Volume Rendering

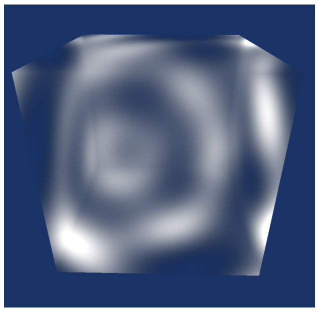
Prerequisites Obtain the skeleton shader (`DVRcomposition.glsl`) from ILIAS and familiarize yourself with ShaderToy (`https://www.shadertoy.com/new/`). ShaderToy only needs a WebGL-compatible browser, such as Google Chrome, Mozilla Firefox, or Microsoft Edge. To get started, just copy the skeleton shader into the source text box on the right and press the play button on the lower left of the source box.

ShaderToy essentially gives you a pre-configured fragment shader you can work with. Opposed to standard fragment shaders from OpenGL, for example, some additional input parameters like the current time, mouse coordinates, or sound processing capabilities are available.

Tasks In the lecture you have learned about different composition schemes for direct volume rendering. Your task is to implement *accumulation (front-to-back)*, *maximum intensity projection*, and *average intensity* compositing schemes.

(a) Accumulation

(b) Max. intensity projection



(c) Average intensity

For front-to-back compositing, you have to walk along the ray while sampling and continuously integrating the volume:

$$\begin{aligned} C^{i+1} &= C^i + (1-\alpha^i) C^{cur} \alpha^{cur} \\ \alpha^{i+1} &= \alpha^i + (1-\alpha^i) \alpha^{cur} \end{aligned}$$

For maximum intensity projection, you have to search for the maximum along the ray. Similarly, for average intensity compositing, you have compute the average value along the ray.

The skeleton shader `DVRcomposition.glsl` already contains a raycasting-based volume renderer. The parameter `technique` controls which composition scheme is used: 0 for accumulation, 1 for maximum intensity projection, and 2 for average intensity composition. Your task is to fill in the code for compositing and take care of the `TODO` comments.

Please test your code using ShaderToy. Then, submit your solution code along with screenshots!

Submission Deadline: 2019-06-28, 23:55

please hand in your submission through the ILIAS system.

.