Assignment 5 - Volume Rendering Using Ray-casting

NAME: SU'AN XIA
STUDENT NUMBER: 18047482
EMAIL: XIASA@SHANGHAITECH.EDU.CN

1 INTRODUCTION

1.1 pre-processing

Requirement: Compute the gradient of the volume data (density).

1.2 Sampler

Requirement: Sample a data point along the ray direction.

1.3 Interpolator

Requirement: Interpolate data from nearby grid points to decide the value of sampled point.

1.4 Classifier

Requirement: Classifier decides its color and opacity based on its values. .

1.5 Compositor

Requirement: Collect and integrate the color and opacity of sampled point one by one.

1.6 Render

Requirement: Assembles all the components above and outputs the resulting image..

2 IMPLEMENTATION DETAILS

2.1 pre-processing

In volume.cpp, we need to complete the function 'void Volume::computeGradients()'.
Use central difference to compute the gradients.
For boundary points, the gradient is the same as the nearest inner point.
For inner points, calculate the corresponding difference in density along dx,dy,dz direction.

2.2 Sampler

In renderer.cpp, we should implement the sample when we assemble all the steps.
A fixed step is ok.

2.3 Interpolator

In interpolator.h, we need to complete the class 'TrilinearInterpolator'
Given the volex,we can easily calculate the density and gradient by formular.

2.4 Classifier

In classifier.h, we need to complete the function 'transfere()'.
We use 'tinycolormap' to map a density to a color.
The we use Phong shading model to render.
Get the final color with considering the effect of gradient.
Get the transparency.

2.5 Compositor

In compositor.cpp, we need to complete the function 'compositeFrontToBack'.
$C_{dst} = C_{dst} + (1 - \alpha_{dst})C_{src}$.
$\alpha_{dst} = \alpha_{dst} + (1 - \alpha_{dst})\alpha_{src}$.

2.6 Render

In renderer.cpp, we need to complete the function 'renderFrontToBack()'.
Assemble all the functions in the following sequences:
Sample, Interpolator, Classfier, Compositor.

3 RESULTS

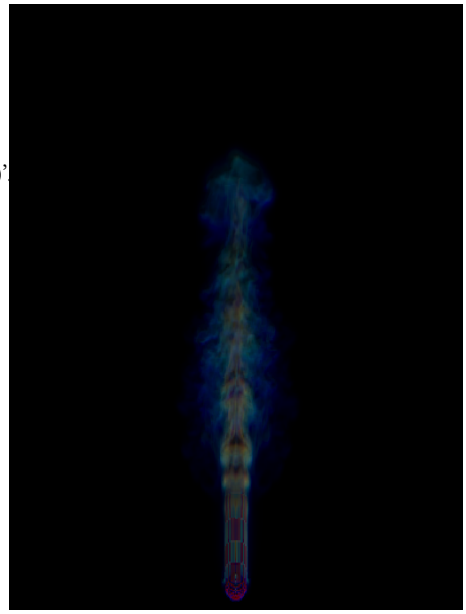

Fig. 1. Nearest,2*dx

1:2 • Name: Su'an Xia
student number: 18047482
email: xiasa@shanghaitech.edu.cn

Fig. 2. Nearest, 1*dx

Fig. 3. Nearest, 0.5*dx

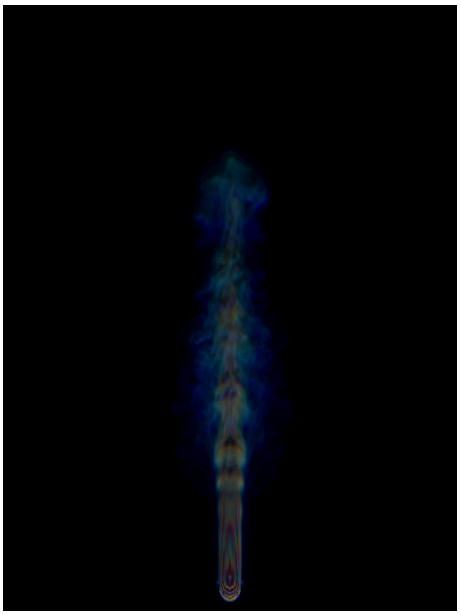

Fig. 4. Trilinear, 2*dx

Fig. 5. Trilinear, 1*dx

Fig. 6. Trilinear, 0.5*dx

Fig. 7. Trilinear, dx/3