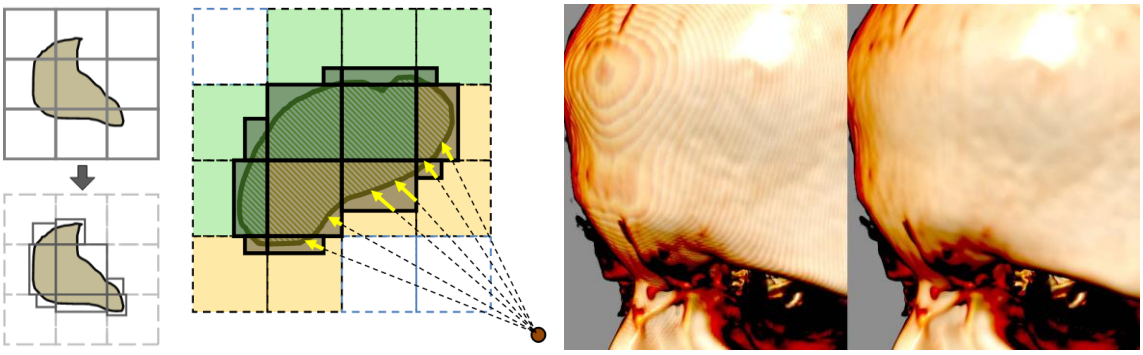
Advanced GPU-based Ray Casting for Bricked Datasets

Nikolay Gavrilov, Vadim Turlapov

Lobachevsky State University of Nizhni Novgorod, Russia



(a) (b) (b)

Figure 1: a) bounding boxes fitting; b) empty space skipping and early ray termination; c) tri-linear (left) & tri-cubic (right) interpolation modes side-by-side comparison (MANIX test dataset).

1. Introduction

Since the 90s, the Direct Volume Rendering shows itself as an efficient tool for the visual analysis of volumetric datasets. There are approaches that allow for the real-time Ray Casting for visualization of the datasets that can be divided into bricks and entirely loaded into the GPU memory. We describe particular details, implemented in our volume rendering engine, that significantly improve rendering quality and performance.

2. Implementation Aspects

In order to be able to visualize the datasets, that do not fit into a single 3D texture memory, we make the data decomposition into bricks. In order to perform correct splicing without any artifacts on the bricks' bounds we perform the bricks' 2-voxel overlapping (3-voxel in tri-cubic filtering case). To render the dataset we render each brick one after another. The bricking allows us to ignore fully empty bricks and do not upload them to the GPU.

Using small 3D textures augments the sampling speed from them during the ray casting process. The front-to-back rendering order of the bricks allows us to ignore those image regions of the bricks that are occluded from the observer by the previously rendered bricks, so that we obtain a '*high-level' early ray termination* (Figure 1b). The *empty space skipping* strategy is also well applicable for the decomposed data because we can bound each single brick separately, which is much easier than to bound the whole dataset (Vincent V. et al, 2008). For each brick we use a simple bounding box, which is fitted automatically to the visible data while the user is applying the transfer function (Figure 1a).

Our engine also provides high rendering quality by reducing certain artifacts. Because of the finite number of the steps the ray may skip meaningful features in the dataset. As the result 'wood-like' image artifacts may appear. This wood-likeness appears because rays start from the same plane (i.e. bounding box face). This artifacts' regularity can be removed by *randomization of ray start positions*. Then these obtained 'randomized' images can be accumulated so that a user will see the average image without noise. The visualization quality can be improved by the *tri-cubic filtering* instead of the common tri-linear one. In order to make a single tri-cubic sampling it is necessary to make 8 tri-linear samplings from the same dataset (Daniel R. et al, 2008).

The optimal brick sizes appeared to be 64^3 and 128^3 cubes depending on the dataset. High performance is mostly caused by the empty space skipping and the small bricks' size. However, the choice of a too small brick size (e.g. of size 32^3) involves the enormous amount of the bricks to draw, which obviously causes the lack of rendering performance. The number of samplings per step is not so crucial aspect for the rendering performance, e.g. the simple DVR is 3 times faster than DVR with tri-cubic filtering, but not 8 times. The rendering time is also wasted when switching to next brick, because we are to copy the rendered result to the OpenGL Frame Buffer for reading. We use this buffer to merge the rendered bricks and to perform possible early ray termination before casting a ray.

References

Daniel R. et al, 2008. Efficient GPU-Based Texture Interpolation using Uniform B-Splines, *In IEEE Transactions on Journal of Graphics, GPU, & Game Tools*, Vol. 13, No. 4, pp 61-69.

Daniel R. et al, 2006, Optimizing GPU Volume Rendering, *Li Journal of WSCG'06 14(1-3),* pp. 9-16.

Guthe, S. et al, 2002, Efficient Interactive rendering of large volume data sets, *VIS 2002*. IEEE, pp. 53 - 60

Johanna B. et al, Smooth Mixed-Resolution GPU Volume Rendering, *IEEE International Symposium on Volume and Point-Based Graphics (VG '08);* 2008. pp. 163–170.

Vincent V. et al, 2008, Simple Empty-Space Removal for Interactive Volume Rendering, *Journal of Graphics, GPU, and Game Tools*, Volume 13, Issue 2, pp. 21-36

SIGGRAPH 2012, Los Angeles, California, August 5 – 9, 2012.
ISBN 978-1-4503-1435-0/12/0008