Workshop on Visualisation in Environmental Sciences (EnvirVis) (2013)
O. Kolditz, K. Rink and G. Scheuermann (Editors)

Rendering technique of multi-layered domain boundaries and its application to fluid flow in porous media visualizations

D. Naumov[†1] and L. Bilke[1] and O. Kolditz[1]

[1]Helmholtz Centre for Environmental Research UFZ, Leipzig, Germany

Abstract

Current visualization techniques for computational fluid dynamics applications are sofisticated and work well in simple geometries. For complex geometries like pore spaces, multiple domain boundaries are obstructing the view and make the studying of fluid flow fields difficult. To overcome these deficiencies we use two-sided materials to render the domain boundaries.
Using this technique it is possible to place the camera inside the domain and to have a non-obstructed view on the surrounding flow field without losing spatial reference to the domain boundaries. As a result a larger part of fluid flow visualization is visible.
Two-sided material rendering was successfully applied to display still images with Blender Cycles renderer and in a virtual reality environment.

Categories and Subject Descriptors (according to ACM CCS): H.5.1 [Information Interfaces and Presentation]: Multimedia Information Systems—Artificial, augmented, and virtual realities I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Virtual reality I.3.3 [Computer Graphics]: Picture/Image Generation—Display algorithms

1. Introduction

There are many fluid flow or particle cloud visualization techniques giving impressive results when used in simple geometries, *i.e.* when the domain boundaries are not occluding the view (see [MLP*10] and references therein for a recent overview of visualization techniques). In the context of fluid flow simulations in random heterogeneous materials (like sandstones or packed beds), the fluid domain boundaries are not simple and the view inside of such a structure is obstructed. The obvious solution to make all of the boundaries transparent or semi-transparent does not solve the issue because the spatial reference to the domain's surface and its influence on the *local* fluid flow field is lost (in the transparent case) or difficult to imagine.

We apply a rendering technique to fluid flow visualizations in pore space geometries to gain deeper insights into the microscopic behaviour of fluid and particles in pore space. The technique is not limited to pore space geometries and is applicable to other visualizations with multi-layered domain boundaries.

In the following, we apply a two-sided material shading technique to fluid flow visualization in a packed bed domain. We also advocate the use of high-quality rendering software for all images because of improved depth perception through realistic shading and shadow mapping algorithms.

Before describing the two-sided material shading application, we briefly introduce the software used in this work to proceed from data input to final visualization. Next, we explain the idea of two-sided material shading followed by its realization using Blender [Ble13] and as well as in the visualization facility TESSIN VISLab [Zeh12].

2. Software and data processing

OpenFOAM for fluid flow simulations We solve the Navier-Stokes Equations for an incompressible Newtonian fluid in a pore space geometry. We use the OpenFOAM computational fluid dynamics solver [Ope13] `icoFoam` to

† dmitri.naumov@ufz.de

DOI: 10.2312/PE.EnvirVis.EnvirVis13.043-046

compute a laminar, steady state solution. The pore space geometry has been derived from a CT-scan of an experimental setup described in [BLK*12, BKL*13] which is a packed bed of spherical pebbles. Their diameter is approximately 0.17 length units in the non-dimensional setup. We set the inlet velocity and the dynamic viscosity to unity.

Particle tracking We use an OpenFOAM solver called `icoUncoupledKinematicParcelFoam`. The particle cloud is injected at the inlet with a constant rate of 10^6 particles per second. The particles are modeled as spheres of a constant size (diameter 10^{-6} length units), not interacting with each other but sticking to the fluid domain's boundaries.

File formats We use the Visualization Toolkit VTK [SML03] as intermediate format for post-processing. OpenFOAM simulation results are converted to native VTK files or imported by Paraview (which is based on the VTK; [Par13]). Files in the native VTK format are easily manipulated either via a Python (version 2) interface to VTK or by using Paraview. The processed data can then be exported into another 3D data format or an image.

Rendering with Blender Cycles Blender is a computer graphics software to create and postprocess 3D data and is usually used by 3D-artists [Ble13]. We use its powerful scene manipulation tools to setup lights and cameras and its interface to several rendering engines to create final images. The current Blender version is 2.66 with Blender Cycles as a rendering engine [Ble12].

Virtual reality environment To study complex models, we are using an interactive visualization environment shown in Figure 2 at the Visualization Center at the Helmholtz Centre for Environmental Research – UFZ. It combines two important features: the stereoscopic view and the interaction with the model being studied. Together with a user tracking system, it creates a feeling of being immersed in the visualization [Zeh12].

The hardware set-up of the TESSIN VISLab uses a back projection-based stereoscopic visualization environment with an approximately 6×3 meter large main screen. Alternating images for the left and the right eye are generated, and users wear special glasses which separate these images, resulting in stereoscopic view. An optical tracking system compensates for observer's movements so that correct perspective is maintained. A pointer device allows for interaction with the virtual environment. The rendering is performed on a cluster with 13 workstations (one for each projector).

3. Two-sided material rendering

Visualization of fluid flow in complex geometries is especially difficult if the fluid domain is hidden by boundaries. Simple boundaries can be rendered semi-transparent or in wire-frame, for example, without losing spatial reference to the domain boundaries. Domain boundaries appearing in multiple layers one after the other, like in heterogeneous porous media, however, occlude the view field and the main fluid flow visualization. Multiple semi-transparent boundaries are difficult to comprehend even in interactive 3D environments with stereoscopic view.

Showing both the fluid flow visualization and all domain boundaries simulataneously without decreasing clarity of the scene is difficult, but in the case of porous media, we are interested in *local* features of the flow. The local features are usually shown similar to Figure 3, where the camera is placed in the void space. In this case, the nearby boundaries hide large parts of the fluid flow visualization (velocity field shown as rendered cone glyphs in this case).

An improved view is shown in Figure 4 in which the camera has been placed near the sphere shown in the middle/bottom of the previous image. Now, the velocity field around all of the nearby spheres is visible. The almost transparent domain boundary has a bluish tint when looking through it (visible on the right-hand side of the image) while the other boundaries are rendered in opaque white.

To create such images, we use a two-sided material, which makes domain boundaries transparent when looking from the inside of the domain (the sphere in the above image) but opaque when looking from the outside.

Realization in Blender and TESSIN VISLab In Blender, two-sided materials are created using the face normal direction $\mathbf{n}$ (together with the camera's pointing vector $\mathbf{c}$). Results of two simple shaders—diffuse and transparent materials—are chosen depending on the sign of the scalar-product between the vectors $\mathbf{n}$ and $\mathbf{c}$ (compare with Figure 1 below): A transparent shader when viewing from the inside $\langle \mathbf{n}, \mathbf{c} \rangle \geq 0$ and a diffuse shader when viewed from the outside $\langle \mathbf{n}, \mathbf{c} \rangle < 0$.

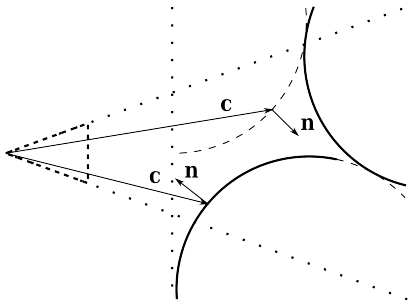

Figure 1: *Two-sided material based on camera's pointing vector* $\mathbf{c}$ *and surface's normal vector* $\mathbf{n}$. *A camera on the left-hand side is shown as triangle with field of view (diagonal dotted lines) and its clipping plane (vertical dotted line). Surfaces (here as circles) are rendered transparent when viewed from inside (dashed line) and opaque when viewed from outside.*

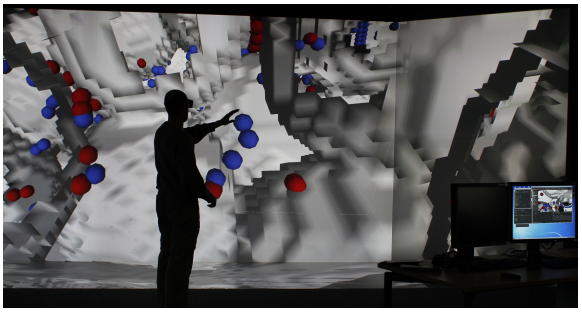

Figure 2: *Interactive visualization in the Visualization Laboratory of domain boundaries and particles sticking to the surface. Using two-sided materials removes the first layer of the surface when looking from inside and allows studing of local properties.*

In the TESSIN VISLab software, we use a feature known as face culling to achieve this effect. In contrast to Blender, the faces viewed from inside are not rendered at all using this software, which can be irritating in an interactive environment. A photograph in Figure 2 shows the same pebble bed dataset. The viewer is looking from inside one of the solid pebbles. Instead of the velocity field, particles rendered as spherical glyphs sticking to the surface are shown.

4. Conclusions

In this work, we have presented a technique for visualizations in pore space geometries. Using two-sided materials, it is possible to retain a non-obscured view on common visualizations in complex domains (like streamlines for fluid flow simulations or glyphs for particle clouds). This technique is especially useful in interactive virtual reality environments where a user controls the camera's location and the viewing direction. It was used in the virtual reality TESSIN VISLab environment. Another application of this technique is to create high-resolution still images using the Blender software.

Using a global illumination renderer with physically correct shader and shadow mapping algorithms in Blender, we rendered high-resolution images showing a fluid flow visualization from the inside of an otherwise opaque domain boundary. Realistic shadowing and rendering of material surfaces helps to retrieve depth information and create volumetric scenes.

The rendering algorithms in Blender Cycles allow creation of almost photorealistic images; correct shading and multiple lights help to position scene elements easier in the 3D space. Near realistic rendering algorithms greatly improve the perceivability of complex scenes at the cost of increased demand for computational resources with higher rendering accuracy.

5. Acknowledgements

The authors would like to thank Thomas Nagel, Jens-Olaf Delfs and Norbert Böttcher, who provided valuable ideas to the writing of this paper, and Leslie Jakobs for proofreading.

The presented work has been partly funded by the German Federal Ministry of Education and Research (BMBF) in as a part of the GEOTECHNOLOGIEN Program. The conceptual development as well as visualization methodologies of this work are also incorporated into the A-DuR (grant ID: 02E10588), CO2BENCH (grant ID: 03G0797D) and INFLUINS projects, funded by BMBF.

6. References

References

[BKL*13] BARTH T., KULENKAMPFF J., LUDWIG M., BRAS S., GRÜNDIG M., FRANKE K., LIPPMANN-PIPKE J., HAMPEL U.: Study of particle deposition and resuspension in pebble beds using positron emission tomography. In *The 15th International Topical Meeting on Nuclear Reactor Thermal Hydraulics (NURETH)* (2013). Unpublished. 2

[Ble12] Blender Release Logs Version 2.61, December 2012. URL: http://www.blender.org/development/release-logs/blender-261/. 2

[Ble13] Blender 3D software, 2013. Version 2.66. URL: http://www.blender.org/. 1, 2

[BLK*12] BARTH T., LUDWIG M., KULENKAMPFF J., GRÜNDIG M., BIEBERLE A., HAMPEL U., LIPPMANN-PIPKE J.: Pet measurements of liquid aerosol particle deposition in pebble beds. In *6th International Topical Meeting on High Temperature Reactor Technology HTR2012* (2012). 2

[MLP*10] MCLOUGHLIN T., LARAMEE R. S., PEIKERT R., POST F. H., CHEN M.: Over two decades of integration-based, geometric flow visualization. In *Computer Graphics Forum* (2010), vol. 29, Wiley Online Library, pp. 1807–1829. doi:10.1111/j.1467-8659.2010.01650.x. 1

[Ope13] OpenFOAM - The Open Source Computational Fluid Dynamics (CFD) Toolbox, 2013. Version 2.1.x. URL: http://www.openfoam.com/. 1

[Par13] ParaView - Open Source Scientific Visualization, 2013. Version 3.98. URL: http://www.paraview.org/. 2

[SML03] SCHROEDER W., MARTIN K., LORENSEN B.: *The Visualization Toolkit: An Object-Oriented Approach To 3D Graphics , Third Edition*. Kitware Inc., 2003. 2

[Zeh12] ZEHNER B.: Scientific 3d visualization—representing complex data sets in a comprehensive way. In *UFZ-Bericht 6/2012* (Leipzig, April 2012), Helmholtz-Zentrum für Umweltforschung GmbH - UFZ, p. 19. 1, 2

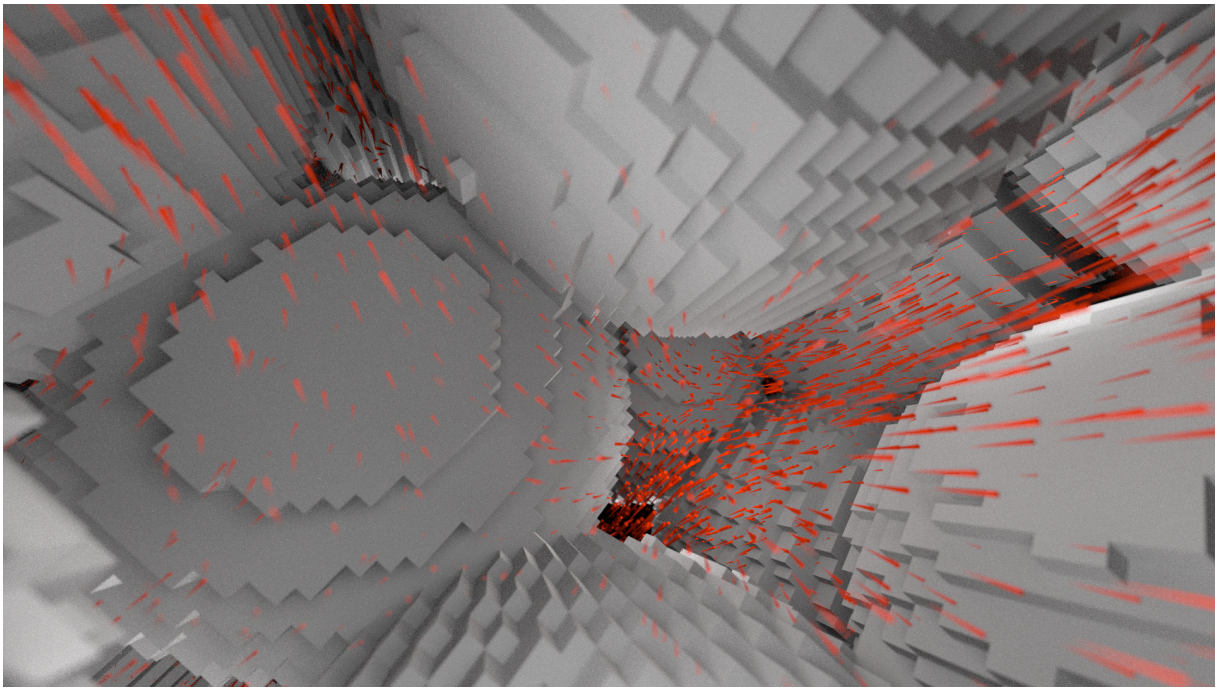

Figure 3: *Rendering of fluid flow velocity field visualization with the camera placed in the void space. Large parts of the velocity field are hidden by the spheres.*

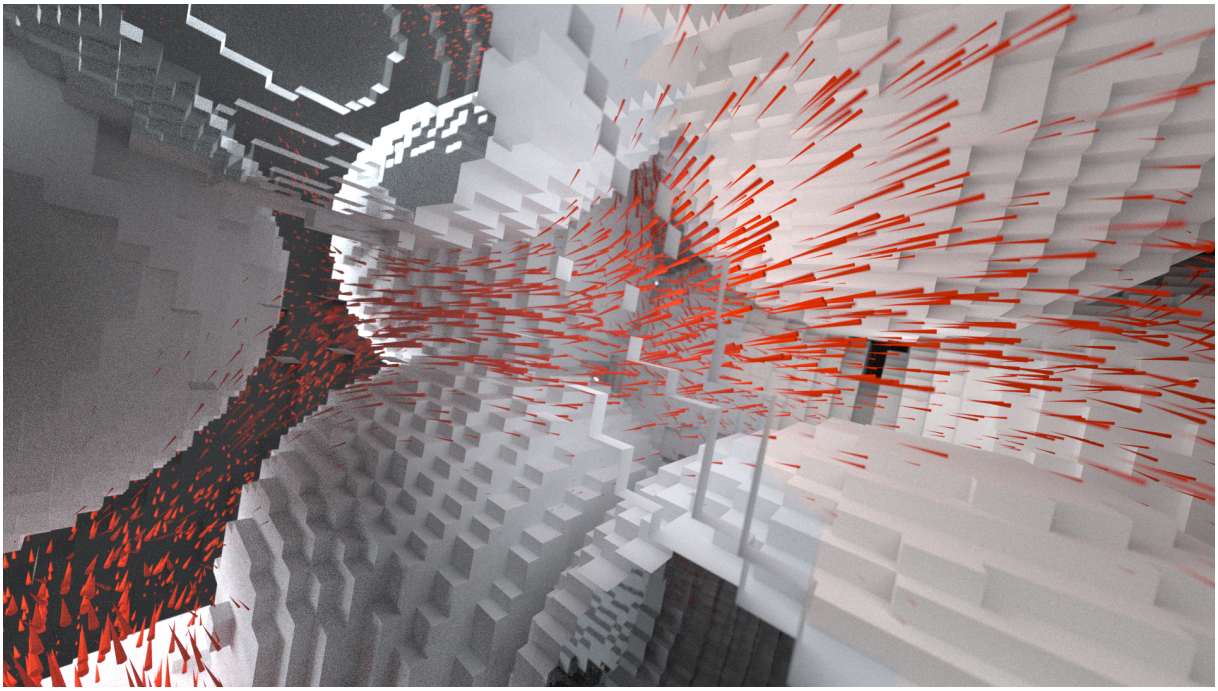

Figure 4: *Rendering of the fluid flow velocity field visualization with the camera placed inside one of the spheres (middle/bottom one shown in previous figure), indicated by the bluish tint. Larger parts of the fluid flow visualization are visible.*