Interactive Visualization of Cerebral Blood Flow for Arteriovenous Malformation Embolisation

Ulrike Sprengel[1], Patrick Saalfeld[1], Sarah Mittenentzwei[1], Moritz Drittel[1], Belal Neyazi[2], Philipp Berg[3,4], Bernhard Preim[1], Sylvia Saalfeld[1,4]

[1]Department for Simulation and Graphics, University of Magdeburg, Germany
[2]University Hospital Magdeburg, Department of Neurosurgery, Germany
[3]Laboratory of Fluid Dynamics and Technical Flows, University of Magdeburg, Germany
[4]Research Campus *STIMULATE*, University of Magdeburg, Germany
sylvia.saalfeld@ovgu.de

Abstract. Arteriovenous malformations in the brain are abnormal connections between cerebral arteries and veins without the capillary system. They might rupture with fatal consequences. Their treatment is highly patient-specific and includes careful analysis of the vessels' configuration. We present an application that visualizes the blood flow after different combinations of blockages of feeder arteries. In order to convey a detailed representation of flow in all regions of the vascular structure, we utilized the *visual effect graph* of the Unity game engine that allows displaying several million particles simultanously. We conducted an informal evaluation with a clinical expert. He rated our application as beneficial in addition to the tools used in clinical practice, since the interactive blockage of arteries provides valuable feedback regarding the influence of the blood flow of the remaining arteries.

1 Introduction

Arteriovenous malformations (AVMs) are vascular malformations in which the blood feeding arteries are directly fused with the veins without an intermediate capillary bed. Their center is called *nidus* and consists of interwoven vessel channels. The channels leading to the nidus arteries are called *feeders*, which are important for treatment since they are supplying the AVM. Cerebral AVMs may rupture and can lead to neurological disorders and epileptic seizures with even fatal consequences [1]. The treatment often comprises a combination of embolization, irradiation or surgical removal.

We describe an application to support a patient-specific planning of cerebral AVM treatment focusing on embolization of the AVM's feeder arteries. In clinical practice, embolization is performed with the aid of a sclerosing agent to close the arteries. Since in general a nidus has several feeders, the order in which these feeder arteries are embolized is highly relevant. For example, the occlusion of one artery can alter the blood flow in the other arteries such that a rupture may be induced [2]. With our application, which represents the different sequence of embolization steps, the treating physician may get additional information about the best possible sequence. For neuroradiology, advances in imaging and computer technology have led to the development of different simulation application including sophisticated virtual reality simulators with haptic feedback, but an intuitive visualization of interactive blood flow in AVM is still missing [3].

In this work, we employ the *Visual Effect Graph* (VFX Graph) of the game engine Unity that is used to display special effects with millions of particles. We use the VFX Graph to create an effective and intuitive blood flow visualization.

2 Materials and Methods

2.1 Medical Image Dataset

For the blood flow visualization, a patient-specific dataset was used (Fig. 1). A cerebral AVM was segmented from a 3D rotational subtraction dataset and the feeder arteries were identified with the treating neurosurgeon. Next, to focus on the relevant vascular sections, we edit the model such that only the feeders and a smaller part of the nidus are contained for the sake of feasibility. In the following, we focus on the three feeding arteries.

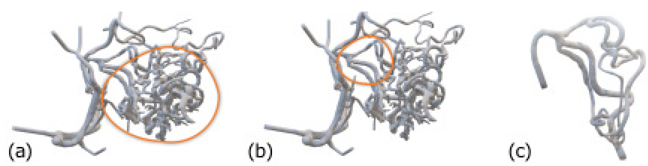


Fig. 1. Depiction of the medical dataset: segmentation of the 3D rotational subtraction data with highlighted nidus (a) and highlighted feeder arteries (b). The model is reduced to focus on these arteries (c).

2.2 Embolization and Blood Flow Simulation

The aim of our AVM blood flow visualization was an intuitive presentation that is easy to understand for medical experts. To ensure a visual representation capable of rendering millions of particles depicting flow patterns even in small regions, we chose the cross-platform game engine Unity (Unity Technologies, Unity v2019.1.3f1, https://unity.com/) together with Unity's VFX Graph. Since the VFX Graph relies heavily on the GPU, an adequate processing of the dataset is required, which is depicted in Fig. 2 and will be described in the following.

Simulation of Embolization. First, we replicated the embolization of the feeder artery, imitating the clinical procedure [1]. A sclerosant consisting of small, often spherical particles, is injected at the root of the artery to prevent disturbances

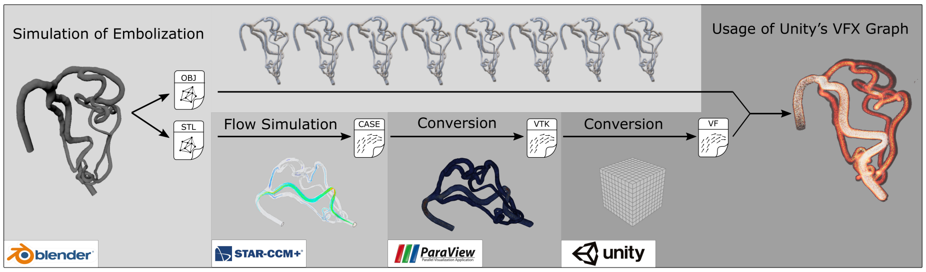


Fig. 2. Depiction of our pipeline containing the workflow from 3D editing, over the flow simulation to the usage of the VFX Graph.

in the blood flow. To analyze if the blood flow simulation can be influenced by a different surface, we tested the simulation with two embolization techniques (Fig. 3). First, we inserted small spheres in the 3D model and second, we cut the feeder arteries such that a gap is created. The 3D models were created with Blender 2.9 (Blender Foundation, Amsterdam, the Netherlands). To enable the selection of the feeders, we prepared all combinations of closed or unaltered feeders, i.e., based on three feeder arteries, we created 2^3 models.

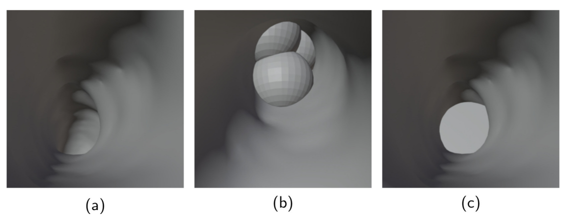


Fig. 3. Depiction of the embolization techniques: (a) unaltered artery, (b) closed artery with spherical particles, (c) closed artery based on cutting.

Blood Flow Simulation. STAR-CCM+ 15.04 (Siemens PLM Software Inc., Plano, TX, USA) was chosen for the blood flow simulation. Based on parameters from the literature [4,5], we set the following attributes: a constant inflow velocity of 0.1 m/s, zero-pressure boundary conditions at all outlets, rigid vessel walls, constant density (1055 kg/m^3) and viscosity (0.004 Pa $*$ s) values. Hence, steady-state conditions and a laminar flow was assumed. We conducted simulations of both closure techniques (recall Fig. 3) yielding no difference regarding the flow behavior. Therefore, we used the plain cuts. Simulation results of each configuration were stored as binary-encoded *.case files.

Usage of Unity's VFX Graph. To use the VFX graph a *VF* file containing a binary 3D texture is required. For this purpose, the binary-encoded *.case files were converted into ASCII-coded VTK files using ParaView (Kitware Inc., New York, USA, https://www.paraview.org/). To convert from VTK to VF files, we mapped the given points on a Cartesian grid using the information provided on the VF file documentation [6]. The step size can be adjusted to influence the spatial resolution. We empirically chose a step size of 0.2 mm.

3 Results

3.1 Interactive Blood Flow Application

Our application was implemented in Unity (Unity Technologies, San Francisco, USA)(Fig. 4). The user can navigate with the keyboard (translate, rotate and scale). In the user interface, she or he can switch between the eight models and adjust the blood flow and the opacity. For the surface visualization of the model, we adapted a ghosted view technique to reveal the underlying particles [7].

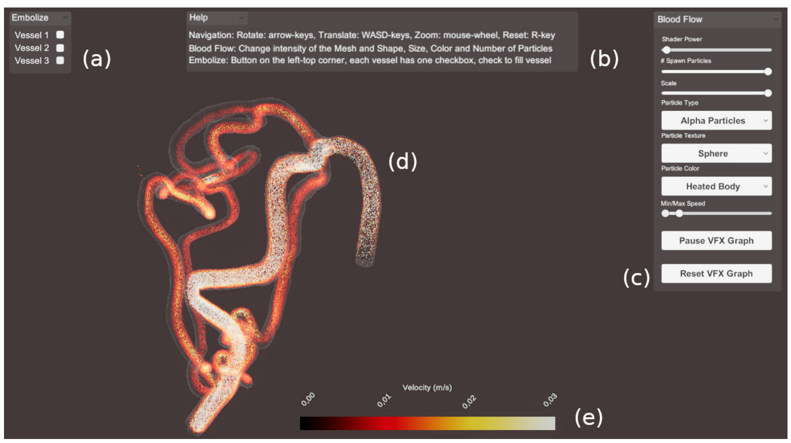


Fig. 4. Depiction of the GUI. On top are the menus for (a) switching between models, (b) usage instructions and (c) adjustable parameters for the visualization. In the center, (d) the surface model and particles as well as (e) the color bar are presented.

The VFX graph has four components: spawn, initialize, update and output. Parameters decide how many particles spawn and simultaneously exist, how long a particle survives and the size of the particles. Furthermore, different blend modes (*alpha*/ *additive*) and visualization types (*particle quads*/*particle strips*) can be chosen (Fig. 5). The particles with additive blending offer a better depth perception because the color values of particles lying on top of each other add up. The user can also choose a color scale to represent the velocity.

To display all 2 million particles using particle quads 259.39 MB memory are needed while the particle strips need only 137.32 MB. Both CPU and GPU are mostly running between 100 and 200 FPS. These values were taken on a PC running Windows 10 x64, with a 3.60GHz AMD Ryzen 5 3600 6-core processor and 16 GB of RAM (1600 MHz) using the AMD Radeon RX 480 graphics card.

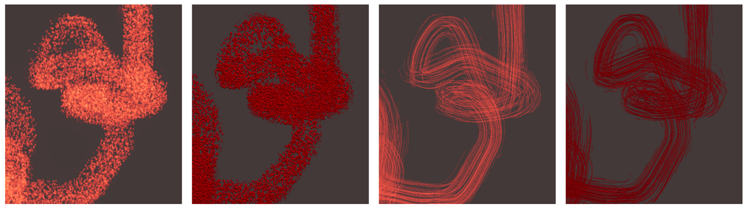


Fig. 5. Available visualization types: (a) particle quads with additive blending, (b) particle quads with alpha blending, (c) particle strips with additive blending and (d) particle strips with alpha blending.

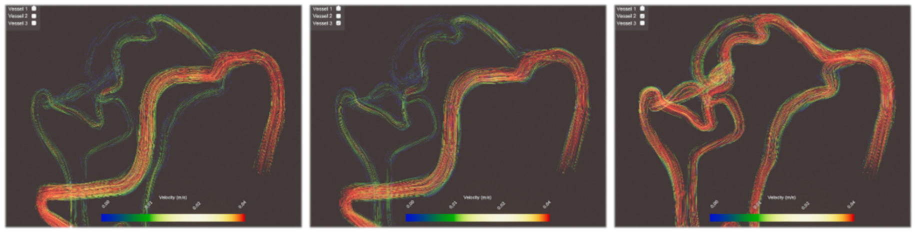
3.2 Evaluation

We evaluated our application via a demonstration session with a neurosurgeon, who is familiar with cerebral AVM treatment. Although he focuses on neurosurgical treatment, he is familiar with embolization therapy as well and interested in the virtual blocking of the different feeder arteries. We used the think-aloud method [8], where the user is encouraged to comment and to provide feedback.

The neurosurgeon liked the visualization especially the possibility to close or open each feeder artery individually. The rainbow-color scale with additive blending was chosen (see Figure 6), but he stated that there is no strong advantage or disadvantage between the scales. Additive blending could better highlight areas with increased blood flow, since the overlaying of multiple particles yields to brighter areas. During evaluation, the visualization of the unaltered arteries indicated that the largest amount of blood flow w.r.t. velocity and particles can be seen in the centered artery, and the artery at the bottom shows the smallest amount. Thus, he would recommend to close first the artery at the bottom, then the artery at top and the artery in the center at last. The neurosurgeon appreciated the visualization of the AVM after the selected blockages and the combination of the eight configurations. For clinical usage, he requested more quantitative information, including wall shear stress as well as volume of the blood flow velocity in the parent vessels compared to the feeder arteries as presented for the characterization of intracranial aneurysm blood flow [9]. He emphasized that a patient-specific planning solely based on the application is not sufficient, but it would be a useful addition for the planning based on pre-surgical datasets, anamnesis and patient's treatment history.

4 Discussion and Outlook

We presented an interactive blood flow visualization of cerebral AVMs. For a representation that allows the visualization of flow patterns even in small regions, we decided to use the game engine Unity including its VFX graph that exploits the graphics hardware to simultaneously render millions of particles. By creating all combinations of feeder artery blockage, the application allows for interactively closing the arteries. A clinical evaluation partner rated it as supportive but in



(a) (b) (c)

Fig. 6. Blood flow visualization with the rainbow-color scale and additive blending. In (a), no artery was blocked, showing largest amount of blood flow for the feeder artery in the center. In (b), the bottom vessel was closed, yielding little changes to the other arteries. In (c) the vessel at the center was closed that inflicts large amounts of blood flow change in the remaining arteries.

addition to the existing clinical practice. Future work should include quantitative blood flow information and should be tested with multiple and more complex malformations. Also the cutout of the vessels should be larger to enable an evaluation of the blood flow within the veins.

Acknowledgements. This work was supported by the Federal Ministry of Education and Research within the Research Campus *STIMULATE* (grant number 13GW0095A).

References

1. Spetzler RF, Martin NA, Carter LP, et al. Surgical management of large AVM's by staged embolization and operative excision. J Neurosurg. 1987;67(1):17–28.
2. Wu EM, El Ahmadieh TY, McDougall CM, et al. Embolization of brain arteriovenous malformations with intent to cure: a systematic review. J Neurosurg. 2019;132(2):388–399.
3. Rehder R, Abd-El-Barr M, Hooten K, et al. The role of simulation in neurosurgery. Child's Nervous System. 2016;32(1):43–54.
4. Ballyk C, Steinman P, Ethier D. Simulation of non-Newtonian blood flow in an end-to-side anastomosis. Biorheology. 1994;31(5):565—-586.
5. Sousa L, Castro C, Conce C, et al. Blood flow simulation and vascular reconstruction. J Biomech. 2012;45(15):2549–2555.
6. Iché T. VectorFieldFile data format; 2020. https://github.com/peeweek/ VectorFieldFile.
7. Behrendt B, Berg P, Beuing O, et al. Explorative blood flow visualization using dynamic line filtering based on surface features. Comput Graph Forum. 2018;37(3):183–194.
8. Van Someren M, Barnard Y, Sandberg J. The think aloud method: a practical approach to modelling cognitive. London: AcademicPress. 1994;.
9. Cebral JR, Mut F, Weir J, et al. Quantitative characterization of the hemodynamic environment in ruptured and unruptured brain aneurysms. American Journal of Neuroradiology. 2011;32(1):145–151.