Visualization of Human-scale Blood Flow Simulation using Intel® OSPRay Studio on SuperMUC-NG

Elisabeth Mayer[1], Jon McCullough[2], Johannes Günther[3], Salvatore Cielo[1] and Peter Coveney[2,4]

[1]*Leibniz Supercomputing Centre, Garching by Munich, Germany*
[2]*University College London, London, United Kingdom*
[3]*Intel Corporation, Germany*
[4]*University of Amsterdam, Amsterdam, Netherlands*

Abstract

Efforts from CompBioMed are directed towards creating a virtual human – a personalized digital representation of a human's biophysical processes. Part of this is HemeLB, a highly scalable, 3D fluid dynamics code that is developing high-resolution simulations of blood flow throughout human-scale vasculature.

Visualizing the results is a significant challenge, particularly due to the complexity and volume of the data. We use Intel® OSPRay Studio for visualizing the resulting data directly on the production machine – LRZ's supercomputer SuperMUC-NG - and producing outputs from 2D to Virtual Reality.

We have created a custom plugin to map the data to the highly memory-efficient VDB volumes of Intel Open VKL for delivering visualization at native resolution and interactive frame rates, eliminating data pre-processing.

In our exploratory video, we outline our approach that allows the full domain simulated by HemeLB to be visualized at a glance, a challenging task with existing techniques.

1. Building a Virtual Human

The successful development and deployment of a virtual human will provide a step change in how healthcare can be tailored to individuals. The concept of a virtual human can be described as a personalized digital representation of a human‘s biophysical processes. It will be able to build on conventional medical imaging techniques to capture the individuality of that person – how their organs, muscles and bones are uniquely structured and interact together.

There are significant healthcare benefits that can be foreseen by the creation of the virtual human. Primary among these is the ability to investigate multiple treatment option and select those that are optimal for the particular individual‘s circumstances. For example, surgical interventions could be pre-planned with unprecedented detail to minimize the chances of relapse, or drugs could be designed and delivery methods planned to reduce side effects. Compared to the well-being of a physical human, repeated studies can be made cheaply on their virtual human representation to ensure that this optimal choice can be made. Such complete integration of numerical simulations into routine healthcare also has the potential to address some of the ethical concerns of medical research. For example the design and prescription of new drug treatments can be examined on a broad range of individuals prior to any human trials taking place. Similarly, the use of numerical techniques can generate quantitative information about physiological processes that can currently only be obtained through highly invasive procedures if at all. The successful development of the virtual human will bring together the fields of medical research, computational science, numerical simulation and data analysis and visualization. Such work is a key area of current research in computational biomedicine worldwide.

2. Simulations, Clinical Imaging and HPC

The development of the virtual human is one of the scientific cases that is helping to motivate the current push towards exascale computing. Capturing the individual nature of human physiology in detail will demand the use of 3D simulation. Conducting human-scale simulations of the separate interacting processes e.g. blood flow through vessels, function of organs, or transfer of nutrients will demand massive computational capacity and generate significant quantities of data. The advance of high performance computing towards the exascale is making the 3D simulation of single aspects of human-scale processes a feasible proposition. Advances in accelerator technology such as GPUs also allow the prospect of simulating relevant timescales in a reasonable wall time.

One area of research that is possibly closest to achieving human-scale simulation is macroscale blood flow through the arteries and veins. To study this, we are using the HemeLB blood flow simulation code. The open source HemeLB code [1] has been developed with a specific focus on the study of three-dimensional vascular flows at high-resolution and at extreme scales. This is enabled through the use of the lattice Boltzmann method (LBM), an alternative approach to solving the Navier-Stokes equations for fluid flow with more traditional finite volume method approaches. The localized computational kernel of the LBM permits two particular advantages for blood flow simulation. The first is that it is a straightforward algorithm to parallelize, leading to good computational performance on both small and large computational scales. The second is that it enables complex geometric shapes to be accommodated using local assignation of the simulation site to its particular category – e.g. fluid, wall, inlet, outlet. HemeLB has been specifically optimized to enable efficient performance on sparse geometries that are characteristic of vascular domains. We have already demonstrated this performance to large scale on some of the largest and fastest supercomputers currently available [3]. Efficient performance to this scale and beyond will be essential for virtual human simulation. Current work is ongoing between project partners to continue to improve the large-scale characteristics of HemeLB.

Effective and efficient simulation of large data sets will be crucial to the success of the virtual human for two key reasons. Firstly, the eventual end-users of virtual human simulations will not be experts in numerical simulation. The outputs must be able to convert simulation data to formats that allow clinicians to easily make decisions in the best interest of the patient and for the patient to fully consent and understand how any planned procedure will impact them. By working with visualization experts, it is possible to create from large data sets simulated on HPC resources, real time experiences. The focus here lays on highlighting the important parts of the data as well as enhance the display of the data to make it readable to the human eye. Visualization plays an important part in science communication and the ability to communicate information through visual mediums. Secondly, 3D exascale simulations will generate equivalent quantities of data. Rough estimates suggest that at least 50 billion data locations will be needed to capture human-scale vasculature at sufficient resolution for accurate simulations. Storing and analyzing this quantity of data remains a subject of significant research and development. Furthermore, big data sets require certain loading times, however, visualizations – especially for science communication – should be in real-time and interactive. For this it is vital to work together with visualization experts to be able to produce an experience that is scientifically correct as well as understandable for a broad audience.

With the dawn of exascale computing fast approaching, and codes such as HemeLB being appropriate analogues for simulation at this scale it is essential to develop effective visualization techniques now. This has been the focus of the collaborative effort behind the current work.

The Intel oneAPI Rendering Toolkit [5] is designed to accelerate scientific visualization and professional rendering workloads with a set of rendering and ray tracing libraries to create high-performance, high-fidelity visual experiences. It consists of several optimized libraries and applications building upon them. We use Intel OSPRay Studio, a lightweight viewer on top of OSPRay [6] with support for many geometry and volume file formats as well as a plugin framework to easily extend functionality and importers. OSPRay uses Intel Embree for building acceleration structures and ray tracing various geometric primitives, Intel Open Volume Kernel Library (Open VKL) to handle and sample different volumetric data sets, and Intel Open Image Denoise to filter Monte Carlo sampling noise from path tracing.

3. Custom HemeLB Importer for Intel OSPRay Studio

The challenge with this project lays within the complexity of the simulated data. The Virtual Human is a simulated version of a patient and will assist clinicians‘ decision making by allowing multiple treatment options to be evaluated on the digital replica of a patient. HemeLB is a 3D blood flow simulation code, capable of simulating human-scale flow models efficiently with the use of HPC. Our workflow has been designed to operate on the supercomputer of the Leibniz Supercomputing Centre (LRZ) [4] – SuperMUC-NG. This is a state of the art HPC resource, ranking 17th in the Top500 list as of July 2021 [9]. Capable of a peak performance of 26.7 PFLOP/s and equipped with 6,480 Lenovo ThinkSystem nodes with Intel Xeon processors (Skylake), 311,040 compute cores and Intel Omni-Path interconnects. HemeLB can run efficiently on this machine and has been shown to scale strongly to the full compute partition [3]. The challenge for HemeLB has been post-processing and the visualization of the the large-scale outputs it is capable of generating. It makes sense for the visualization to take advantage of the HPC environment where HemeLB runs.

So with HemeLB producing large volume data sets consisting of sparse data and creating a unique challenge for visualization and LRZ‘s support not only with HPC resources but also with it‘s team of visualization experts at the Virtual Reality and Visualisation Centre (V2C), who specialize in big data visualizations as well as VR experiences are involved, the next step is what tool to use for the visualization. The tool of choice here being Intel‘s oneAPI Rendering Toolkit, which offers the needed applications for this task.

From the toolkit, the main application used is OSPRay Studio, where a custom HemeLB I/O plugin was created to map the volume data directly into memory with Intel Open Volume Kernel Library. By using Intel OSPRay Studio from a development branch of the oneAPI rendering toolkit, it is possible to use the plugin with native synergies from the studio like the ability to import glTF files as well as animations from, e.g. Blender, to animate cameras, as well as use Intel Open Image Denoise on the images and select various output formats.

The capability of our plugin to directly importing the native, binary output of HemeLB is a particularly useful feature of our workflow. This eliminates the necessity of data pre-processing, which notoriously comes with significant drawback and compromises:

- Waste of computing time (for format conversion) and disk space (for data copies, which sometimes need to be in text format);
- Necessity of forking the workflow towards different software (need for extra knowledge and workforce in the group) and/or hardware (e.g. copying data to visualization servers);
- Eextreme difficulties in automating the visualization, without requiring direct human intervention or supervision;
- Reduction of data resolution (as such hardware often have smaller capacity than where the simulations were run) or anyway manipulation of the raw figures (e.g. re-sampling, conversions from volume to particle data).

As far as we know, this is the first and only visualization workflow allowing direct importing of HemeLB outputs. This matters especially when evaluating the performance (in terms of frames/second), as one cannot really talk of speedups compared to largely manual workflows (or hardly of HPC at all, as the human element is by far the largest bottleneck). Typical previous workflows included conversion of grid outputs to text-based lists of points for importing on software such as VisIt and ParaView [7, 8], only to then need to either manually reconstruct the original grid or using particle-based plotting techniques (thus subject to most drawbacks described above). While it is possible to use OSPRay with similar overall performance for the last steps (as the aforementioned tools come with their own embedded version of it), the pre-processing step can easily consume hundreds of times the wallclock time of the actual rendering. Tests with VisIt (Figure 1) on outputs of 66.4 million elements per time snapshot (about 1.8 GB in memory) on a single SKX node of SuperMUC-NG resulted in rendering at 1080p of order seconds per frame (if underlying data and transfer function do not change), but about 130 seconds of pre-processing steps, not including memory readouts. Our OSPRay Studio plugin with the same conditions and resources produces a frame in about 15 seconds (although a more refined product overall, thus a fair comparison is not possible).

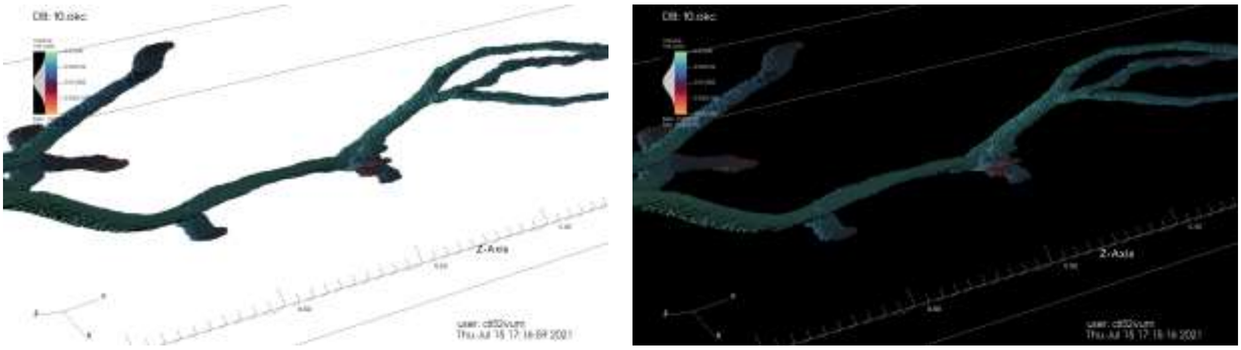

Figure 1. Visualizations of HemeLB data generated using VisIt .

Intel OSPRay Studio offers native functions that are vital for visualizations, e.g. the choice to switch between interactive rendering over the graphical user interface (GUI) or to submit batch renderings that might take up more time. The application also offers the ability to produce not only perspective camera renderings but also panoramic stereo or 3D images in various formats. The images and videos produced can be created by the various renderers of Studio, e.g. SciVis, an efficient ray tracer, or path tracer, which supports various lighting options from ambient, point and spotlight to high dynamic range images (HDRI).

4. Exploratory Visualization in OSPRay Studio

To provide a demonstration case for our workflow, we used HemeLB to generate a large volume data set consisting of sparse data simulation of an arteriovenous fistula. Surgeons can create an access point for hemodialysis by artificially connecting a major artery and vein – an arteriovenous fistula – bypassing vascular beds and increasing flow through the vein. The simulated data in this case includes 166.6 million data points for the veins and 65 million data points for the arteries in 64 timesteps. In total the data is 473GB with 7.4GB per timestep, the simulation of a single heartbeat took 80 minutes on 1,500 nodes of the LRZ‘s SuperMUC-NG.

A real-time exploration of the data using Intel OSPRay Studio on SuperMUC-NG shows, after loading a snapshot of the forearm data, the ability to change the transfer function of the data - in this case increase the opacity, to be able to read the data clearer. Following this, further functions are displayed, such as changing the background color, to raise the contrast between background and data. Exploration of the data set requires a mouse click and drag across the screen, OSPRay Studio adapts the camera perspective relatively quickly.

Further synergies are shown by being able to quickly change between different rendering options (Figure 2) each with individual advantages. For example, the scivis option is a very efficient ray tracer and the path tracer has the ability to display various lighting settings.

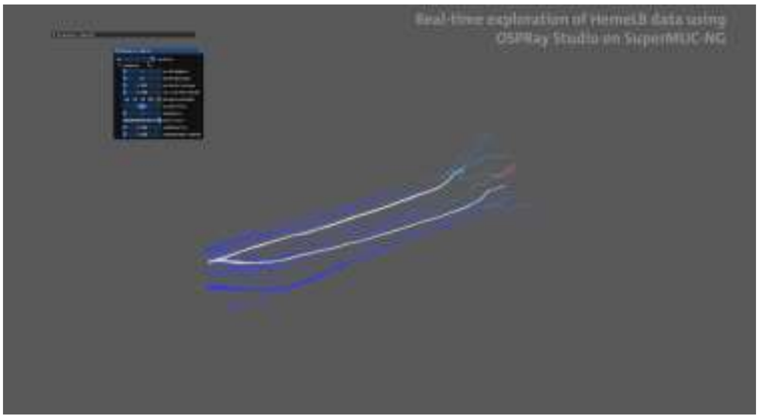

Figure 2. Our Intel OSPRay workflow allows multiple different rendering options to be applied to large and complex datasets.

To show this, there is a switch of lighting from the default ambient light to a high-dynamic-range images (HDRI) commonly used in 3D visualizations to produce a lighting based on an image. Here a simple studio HDRI is used, then its visibility and intensity adapted to the data.

The camera settings options of Intel OSPRay Studio are shown, here the use of depth of field to direct attention onto the arteries in the data set. Through the exploration of the data in almost real-time these settings can be explored and saved for further use with reasonable effort.

Lastly the results show the exported images from Intel OSPRay Studio, ranging from 2D images, which display the pressure in the forearm veins, whilst using the depth of field to focus on the hand region. Followed by 2D videos showing an animation of the pressure in the forearm over the course of 64 timesteps, and then an interactive panoramic capture. With the panoramic images (Figure 3) the first step into developing immersive VR have been taken. Such advances will be essential in making virtual human simulations understandable by end-users such as clinicians and patients.

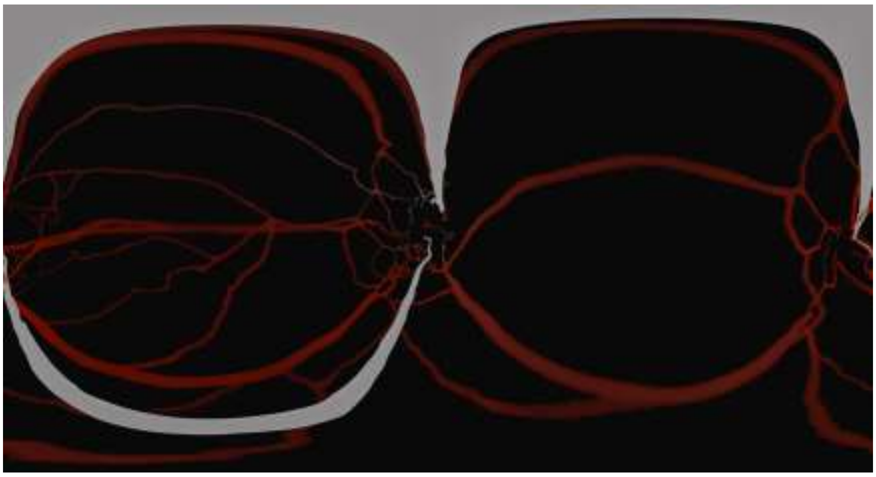

Figure 3. Example of a 360° panoramic rendering from within the HemeLB data. Such images can be used to immerse a viewer within the vasculature using a VR headset.

5. Summary and Outlook

Since the workflow enables the output of various formats (2D to VR), it is possible to view the data using a VR headset from different perspectives. The next step will be to extend this to allow an immersive exploration of the animations. Further with HemeLB, new and even larger data is being produced to examine the scalability of our workflow. For example, we are testing a high-resolution circle of Willis test case that generates over 280GB of data for every output, with 5 outputs that is a total of 1.4TB. This indicates the scale of the virtual human, which shows that data will only become larger in the future, so there will always be a need for efficient visualization methods. To visualize the virtual human, not only veins and arteries will be needed and every organ may have a different data structure and output format. This is the next challenge for visualizing the virtual human and biomedicine. So, extending this method to other research fields and data types is crucial, e.g. astrophysics and geophysics.

Our current workflow is used for exploratory understanding of the large and complex datasets characteristic of human-scale blood flow simulations. It allows the user to efficiently render and interpret fluid flow using the same computational platform on which the simulation was conducted. In the development of virtual human simulations, this will provide a platform for the explanatory visualizations needed for end-users such as clinicians and their patients to understand the treatment options represented by the simulation data and to make treatment decisions.

References

1. HemeLB Website – https://hemelb.org
2. HemeLB Code Repositories – https://github.com/hemelb-codes (main), https://github.com/UCL-CCS/HemePure, https://github.com/UCL-

CCS/HemePure_SelfCoupled

3. J. W. S. McCullough, R. A. Richardson, A. Patronis, R. Halver, R. Marshall, M. Ruefenacht, B. J. N. Wylie, T. Odaker, M. Wiedemann, B. Lloyd, E. Neufeld, G. Sutmann, A. Skjellum, D. Kranzlmüller and P. V. Coveney, "Towards blood flow in the virtual human: efficient self-coupling of HemeLB", *J R Soc Interface Focus*11, 20190119 (2020), DOI:10.1098/rsfs.2019.0119
4. LRZ Website – https://www.lrz.de
5. Intel® oneAPI Rendering Toolkit – https://software.intel.com/content/www/us/en/develop/tools/oneapi/rendering-toolkit.html
6. I. Wald, G. Johnson, J. Amstutz, C. Brownlee, A. Knoll, J. Jeffers, J. Günther, P. Navratil, "OSPRay – A CPU ray tracing framework for scientific visualization", IEEE Transactions on Visualization and Computer Graphics 23 (1) (2017) 931–940. DOIoi:10.1109/TVCG.2016.2599041
7. VisIt Website - https://visit-dav.github.io/visit-website/
8. ParaView Website - https://www.paraview.org/
9. Top500 list June 2021 - https://www.top500.org/lists/top500/list/2021/06/