# An automated pipeline for real time visualisation of blood flow during treatment of intracranial aneurysms

Richardson, R.A.[1], Younes, G.[2], Soheilian, S.[3], Abinahed J.[2], Dakua, S.P.[2], Xiaojun, Z.[3], Chen, M.[3], Amira, A.[3], Zakaria A.[4], Bensaali, F.[3], Balakrishnan S.[2], Baobeid A.[2], Al-Ansari A.[2], Coveney, P.V.[1,5]

[1] Centre for Computational Science, Dept. of Chemistry, UCL, UK
[2] Surgical Research Department, Hamad Medical Corporation, Qatar
[3] College of Engineering, Qatar University, Qatar
[4] Neurosciences Institute, Hamad Medical Corporation, Qatar
[5] Informatics Institute, University of Amsterdam, Netherlands

## 1. Introduction

Imaging and computing technologies have advanced considerably in recent years leading to their increasing use in medical applications. Modern imaging methods allow clinicians to view the geometry of a patient's vasculature, down to the level of individual vessels, allowing vascular malformations such as intracranial aneurysms to be located and examined. On a related front, the increasing availability and computational power of high performance computing (HPC) infrastructure now allows for detailed haemodynamic simulations to be executed [1]. Indeed, advanced software suites have already been developed for hemodynamic simulation from medical imaging, such as CRIMSON [2], including coupling to heart models [3]. The treatment of intracranial aneurysms is often performed under very short timescales. Our aim in this work is to use data which is already available and routinely collected in the process of interventions to treat patient aneurysms - such as rotational angiogram (RA) data - and combine this with high performance haemodynamics simulation codes to provide clinicians with real time visualisation of the predicted blood flow and associated wall shear stresses in the patient before and after the introduction of a flow diverting stent. We present here the fully integrated, automated pipeline we have developed to segment imaging data, localise aneurysms, simulate blood flow and provide real time visualisation to clinicians and some details of its current performance.

## 2. Segmentation of vasculature and localization of aneurysms

Active contour, region-growing and gradient-based methods are among the most preferred segmentation methods, because they are user

friendly, robust to noise, and results oriented. As we are mainly interested in segmenting 3D Rotational Angiography (3DRA) images that typically have high contrast, we have developed a threshold based method using the features of these methods. We have implemented a Slicer module (SegmentVasculature) that takes a 3DRA image as input and outputs a binary/probabilistic segmentation of the vasculature (see Fig. 1A).

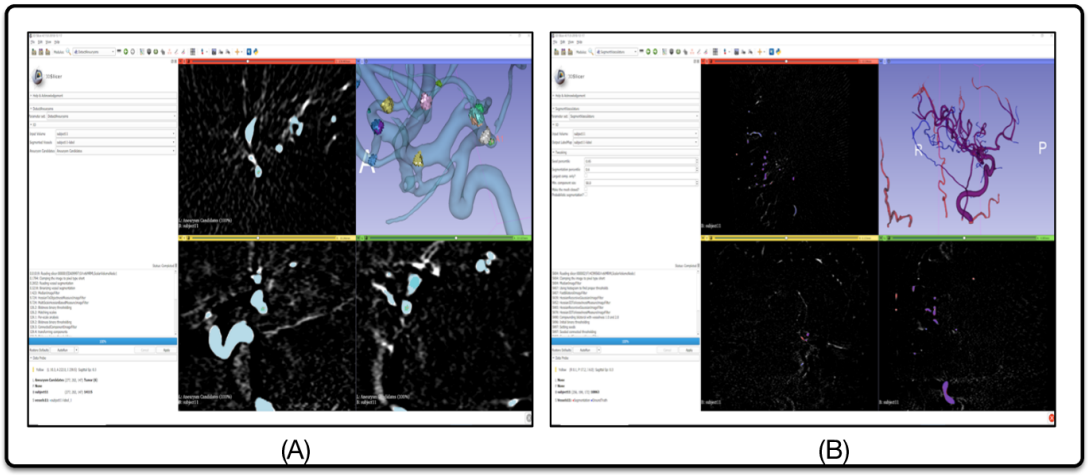


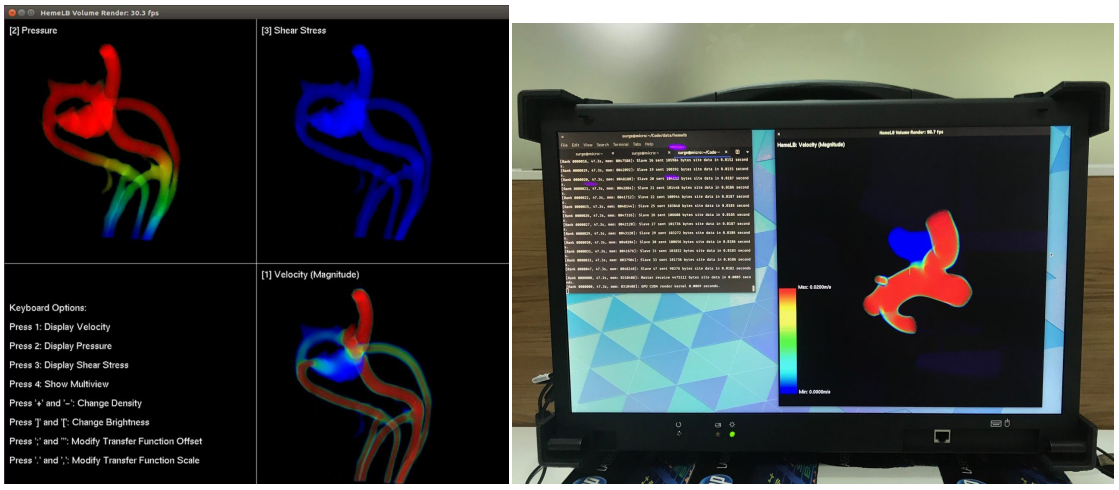

**Figure 1:** ***A:*** *Sample result of the SegmentVasculature CLI on subject11 (red - automated segmentation, blue - ground truth segmentation, purple - overlap).* ***B:*** *Sample result of our aneurysm detection module on the same subject*

We have also developed a method to detect the aneurysm (see Fig. 1B) through scoring of candidate features according to elongation, size and roundness within certain bounds (highest scoring candidate is identified as the aneurysm). We have qualitatively and quantitatively validated both the above methods, obtaining a good Dice coefficient score (above 90%) when testing the segmentation accuracy. The ground truth data were prepared by expert clinicians at Hamad Medical Corporation (HMC). The datasets used to test the proposed system were obtained using Siemens' Artis Axiom Angiography Interventional System.

### 3. Haemodynamic simulation

Creating a clipped 3D surface mesh of the segmented, localised aneurysm region, the holes in the mesh are identified automatically and flow inlets determined from user selection (remaining holes in the mesh are treated as outlets). The creation of appropriate input files for simulation is dependent on the particular computational fluid dynamics package being used - for this work we used HemeLB [4], a high performance lattice-Boltzmann (LB) solver optimized for sparse geometries such as the human vasculature. LB models the fluid region on a regular grid, which

we generate automatically from the surface mesh of the geometry. This regular grid creation has the advantage of being typically less complex and faster than the creation of unstructured meshes necessitated by other approaches such as finite element solvers. Furthermore, LB scales extremely well to very high core counts [5] and as such can more efficiently exploit the increasing availability of massively parallel computing infrastructure to produce results within a clinical time frame.




**(A)** **(B)**

**Figure 2:** ***A:*** *multi-view visualisation window. The 3D display of the results can be illustrated in two different modes: 1) multi-view window which is depicted in this Figure; 2) full separate window for each lattice property. The multi-view window splits into 4 sub windows; one for the visualization instruction as shown in the bottom left and three for the velocity, pressure and shear stress.* ***B:*** *The portable workstation running part of the pipeline.*

## 4. Real time visualisation of results and clinical report

The visualisation of HemeLB is implemented directly on dedicated GPU cores with the support of the computer unified device architecture (CUDA) library. The existing message passing implementation (MPI) method in HemeLB has been used to facilitate the communication between the visualisation client and the rest of LB compute nodes. The compute nodes mainly process the lattice-Boltzmann operations, and transfer the lattice properties caches to the visualisation client, where the visualisation node will share the relevant data with the GPU nodes for rendering. The CUDA cores execute the ray casting kernel function as a direct volume rendering of the lattice properties. In Fig. 2A, some important flow properties (pressure, shear stress, and velocity) are visualised in the same window simultaneously. A set of keyboard options can be then selected for enabling the rotation of each sub window to view

the results from different angles. The dynamic colour map has been set based on the minimum and maximum unit of each simulation result.

## 5. Pipeline performance in clinical context

For testing within the clinic, the pipeline was implemented on a standalone, portable workstation with an integrated Full HD IPS display (see Fig. 2B), running Fedora 30 Workstation Edition. The workstation has two Intel Xeon E5-2680 @ 2.50GHz processors with 24 cores each (48 cores total) and an NVIDIA Quadro K5200 GPU. The approximate time taken for segmentation is 1.5 minutes, and 2 minutes for aneurysm localisation. An example geometry (subject11) with around 186000 fluid sites on 48 threads can run at 741 LB updates per second. Stress testing with GPU rendering every 3 steps decreases this drastically to 59 updates per second. For faster non-blocking rendering, and due to the need for higher resolution simulations, the workstation can connect to the RAAD2 supercomputer in Doha for job submission, with rendering performed on the local workstation. With the small test case mentioned above, we achieve 431 updates per second on the same number of cores, and rendering at around 4 to 8 frames per second, currently limited by the file transfer between RAAD2 and the local workstation.

## 6. Conclusion

We have provided a very brief overview of the three stages in our automated pipeline intended for use by interventional radiologists during treatment of intracranial aneurysms. Future work will focus on the optimization of the pipeline's performance, real time steering of the simulation, and further consultation with clinicians to identify more useful metrics and display strategies. All potential patient recruits were screened for eligibility based on the protocol of the study. All interventions for the project were subjected to Hamad Medical Corporation's institutional review board (IRB) policies, approval and monitoring.

## 7. References

[1] Groen, D. et al. (2018). *Front. Physiol. 9: 721.*
[2] Khlebnikov, R. et al. (2015) *Workshop on Clinical Image-Based Procedures* (pp. 10-18). Springer.
[3] Arthurs, C. J. (2016). *Comp. Biomech. Med.* (pp. 155-166). Springer.
[4] Mazzeo, M. D. et al. (2008). *Comp. Phys. Comm.*, *178*(12), 894-914.
[5] Patronis, A. et al. (2018). *Front. physiol.*, *9*, 331.