Efficient Visualization of Salient Manifolds in Scalar, Vector, and Tensor Fields

Samer Barakat, Xavier Tricoche

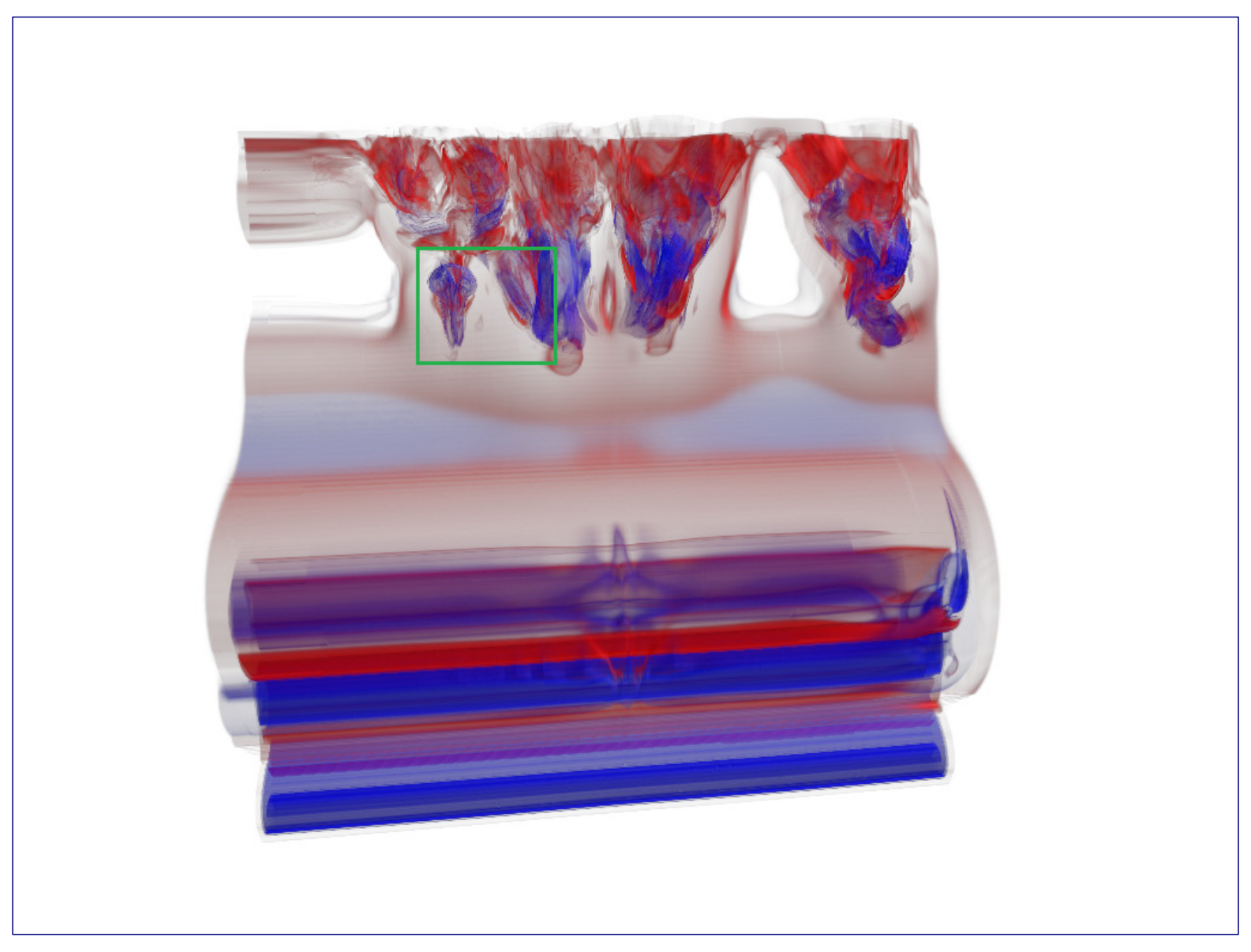
Department of Computer Science, Purdue University

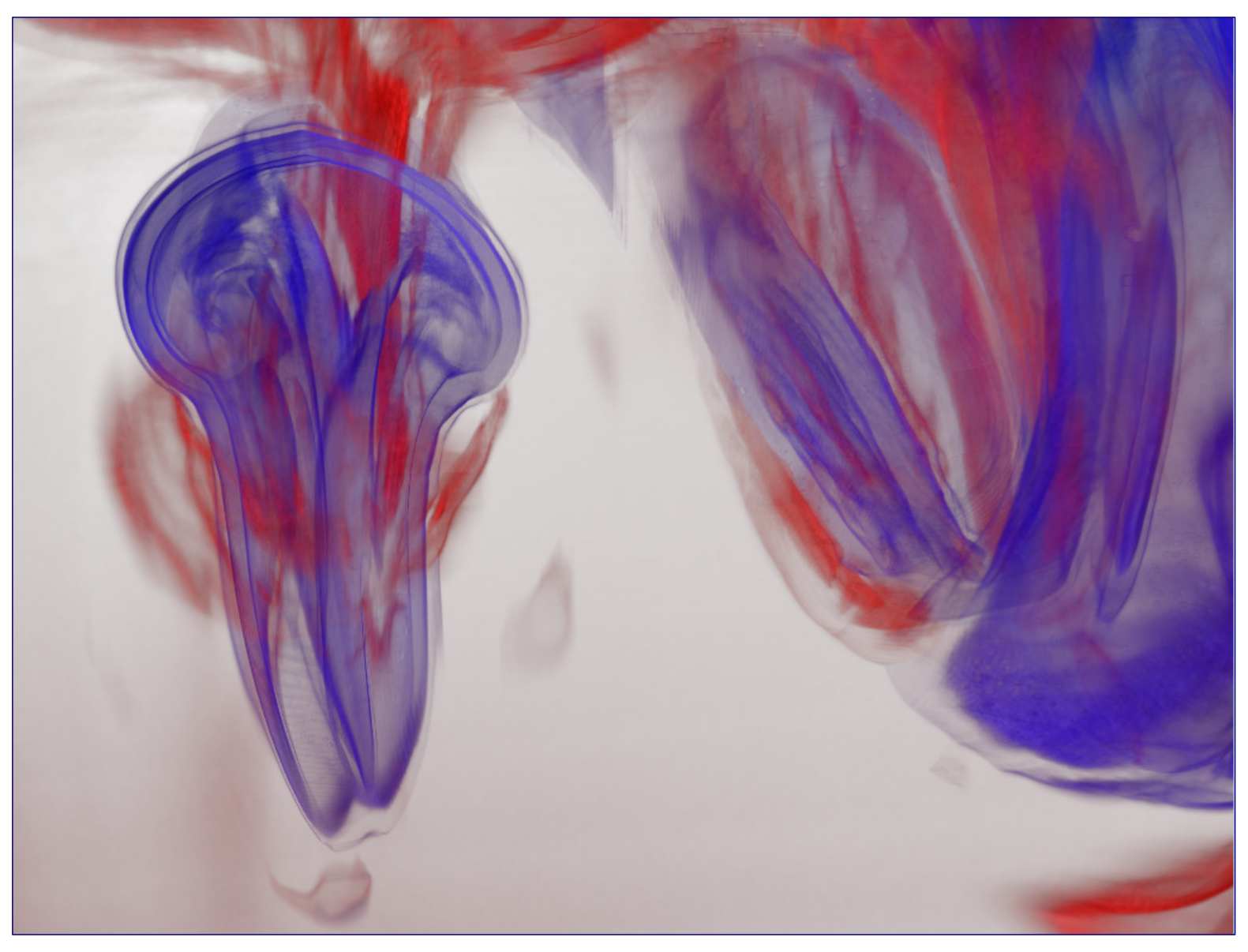
Introduction:

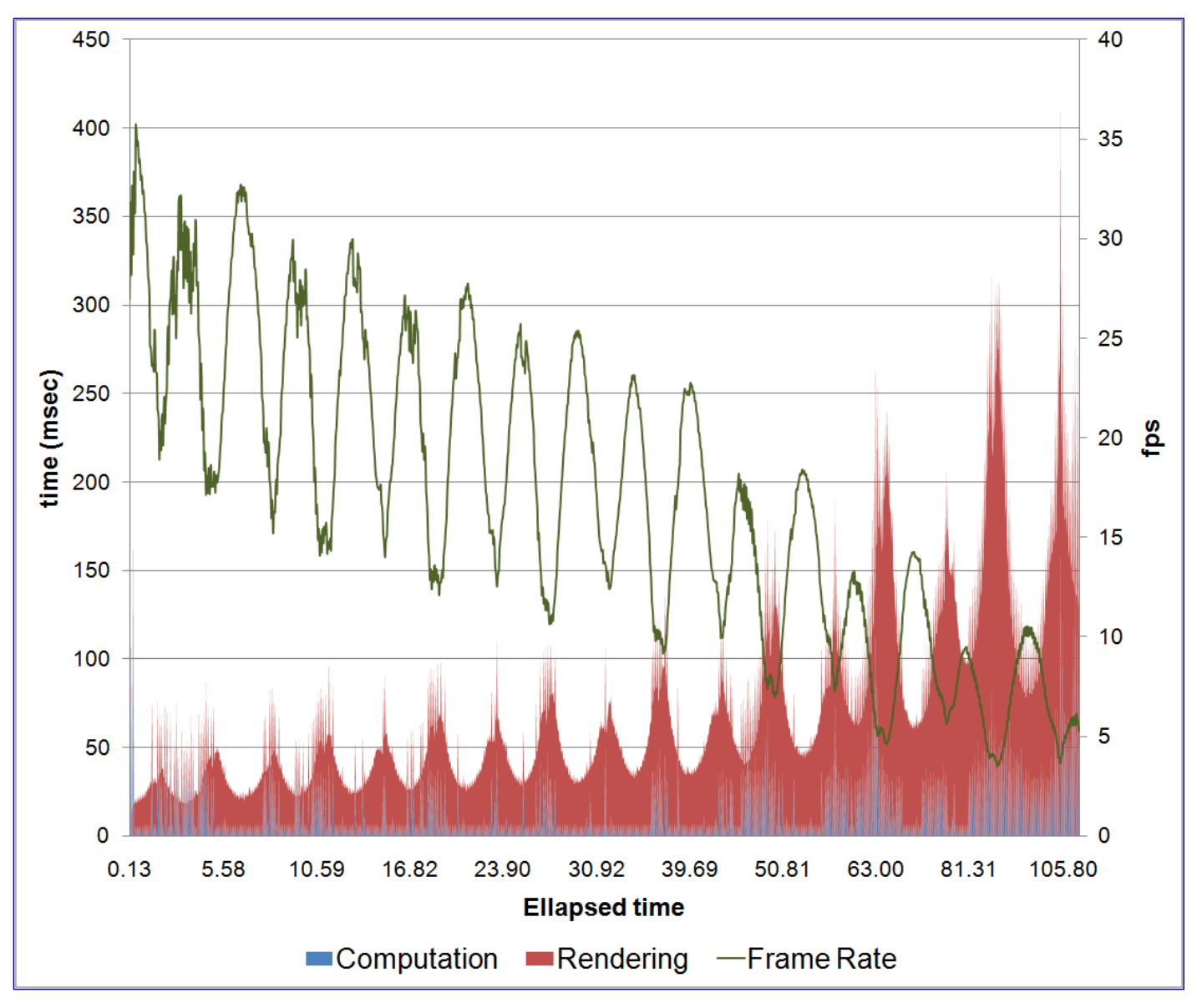
Our research focuses on harnessing the massively parallel compute power of the GPU to visually explore complex datasets. We propose adaptive GPU-based approaches that intertwine computation and rendering. To that end we present novel dynamic data structures for the GPU. Our work includes the visualization of salient structures in vector fields using so-called Lagrangian coherent structures, extraction of ridge and valley manifolds in scale space from volumetric multivariate fields, and efficient and high-quality volume / surface rendering.

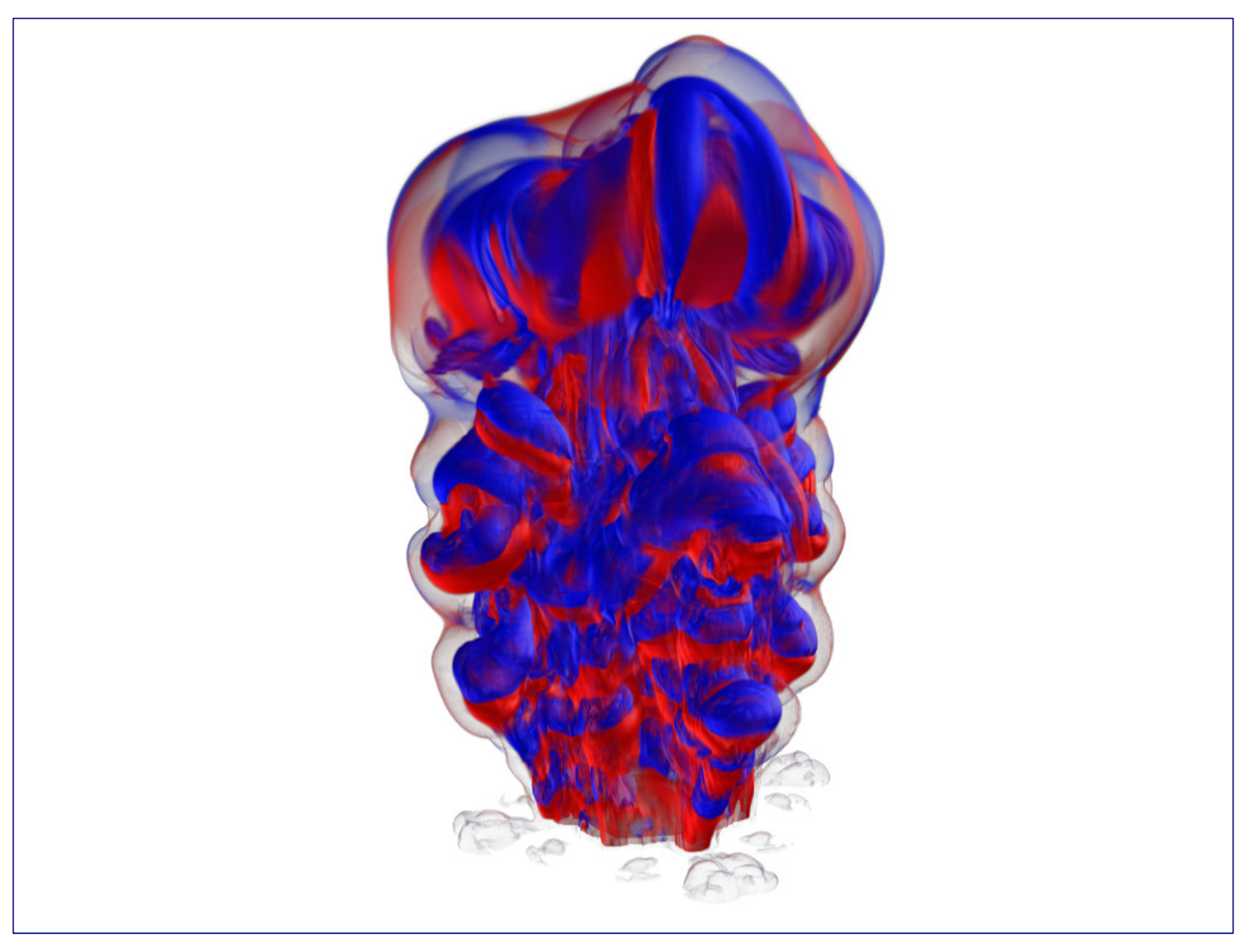
Visualization of Lagrangian coherent structures (LCS)

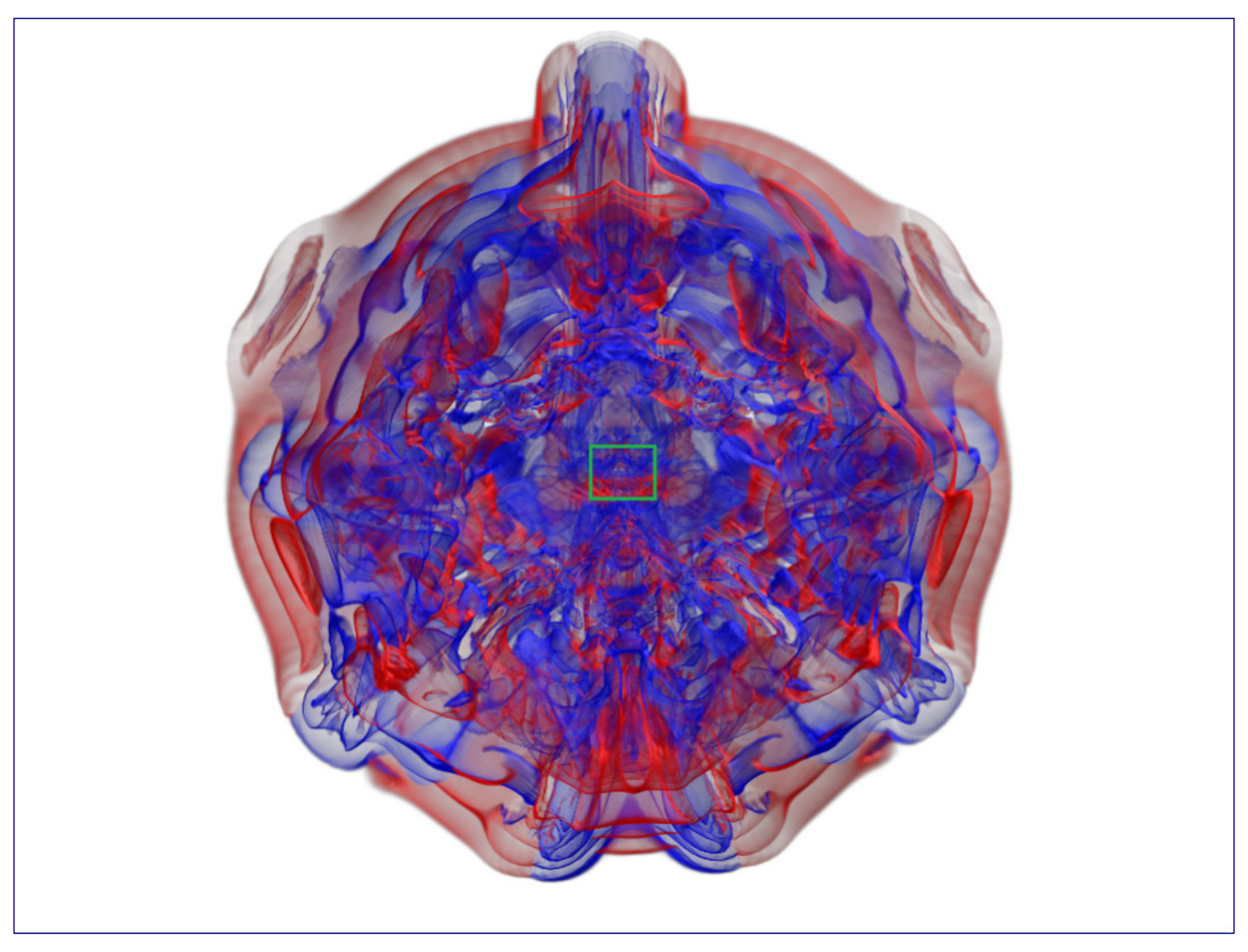
LCS visualization is essential to understand salient patterns in transient fluid flows (e.g. propagation of an oil spill in the Gulf). The characterization of LCS requires the compute-intensive advection of particles (mapped to GPU light weight threads) along the considered flow. Traditionally this computation is carried out everywhere in the domain and at a resolution that can widely exceed that of the input data itself (See figures of consecutive zooming levels). We instead carry out computation and rendering in a view-dependent fashion. At each frame we decide what needs to be computed based on a priority metric that factors in the current visual setting and the exploratory behavior of the user.

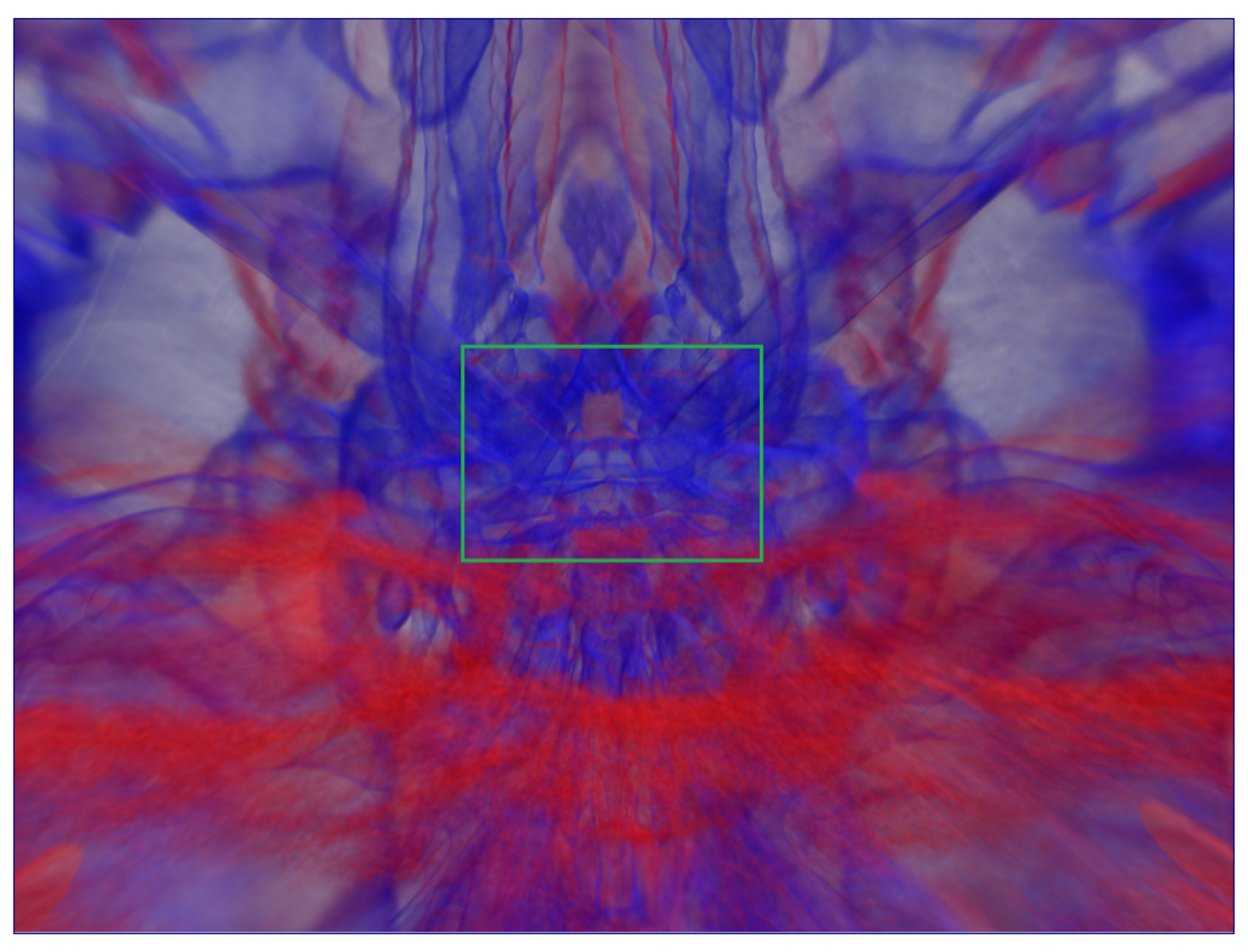


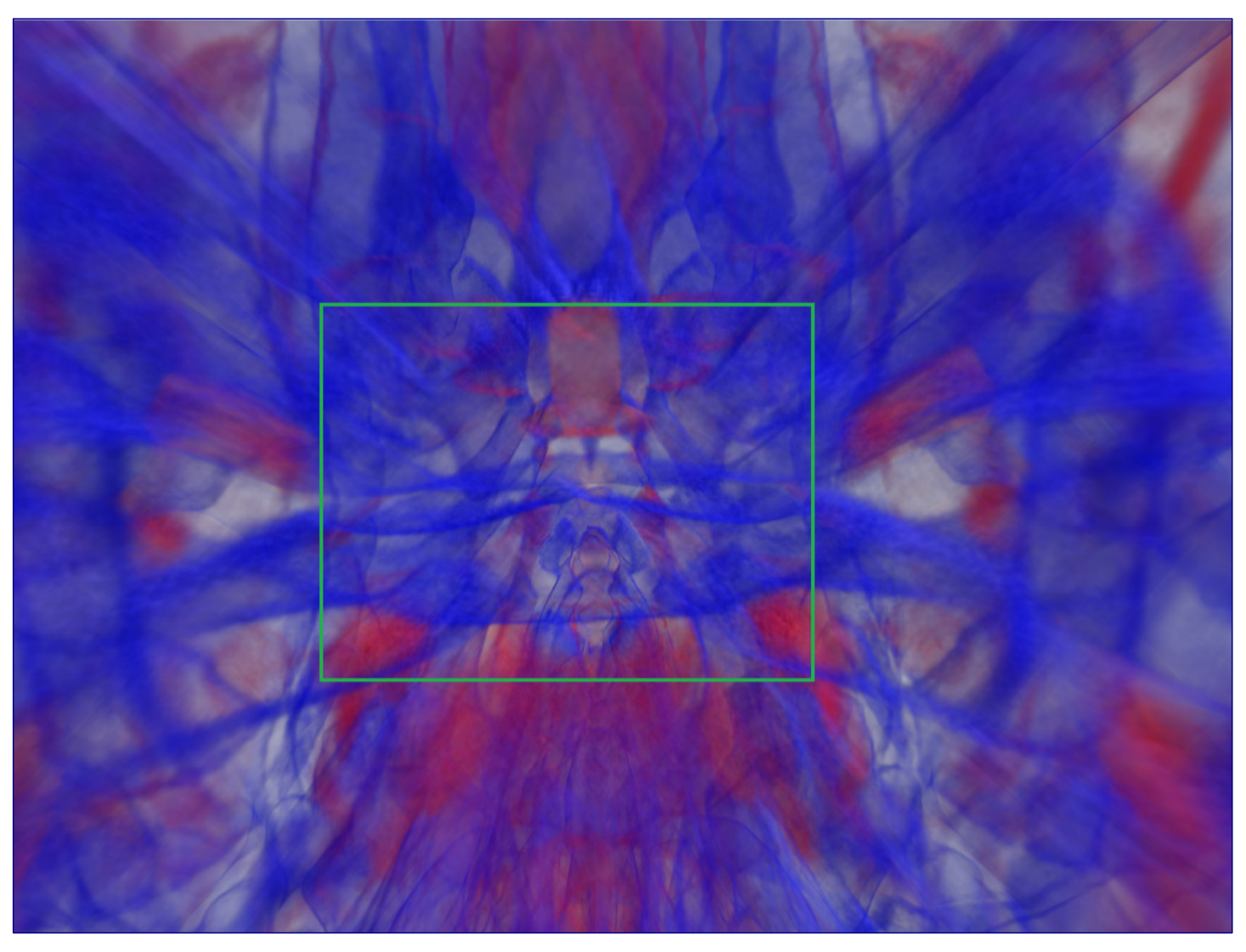


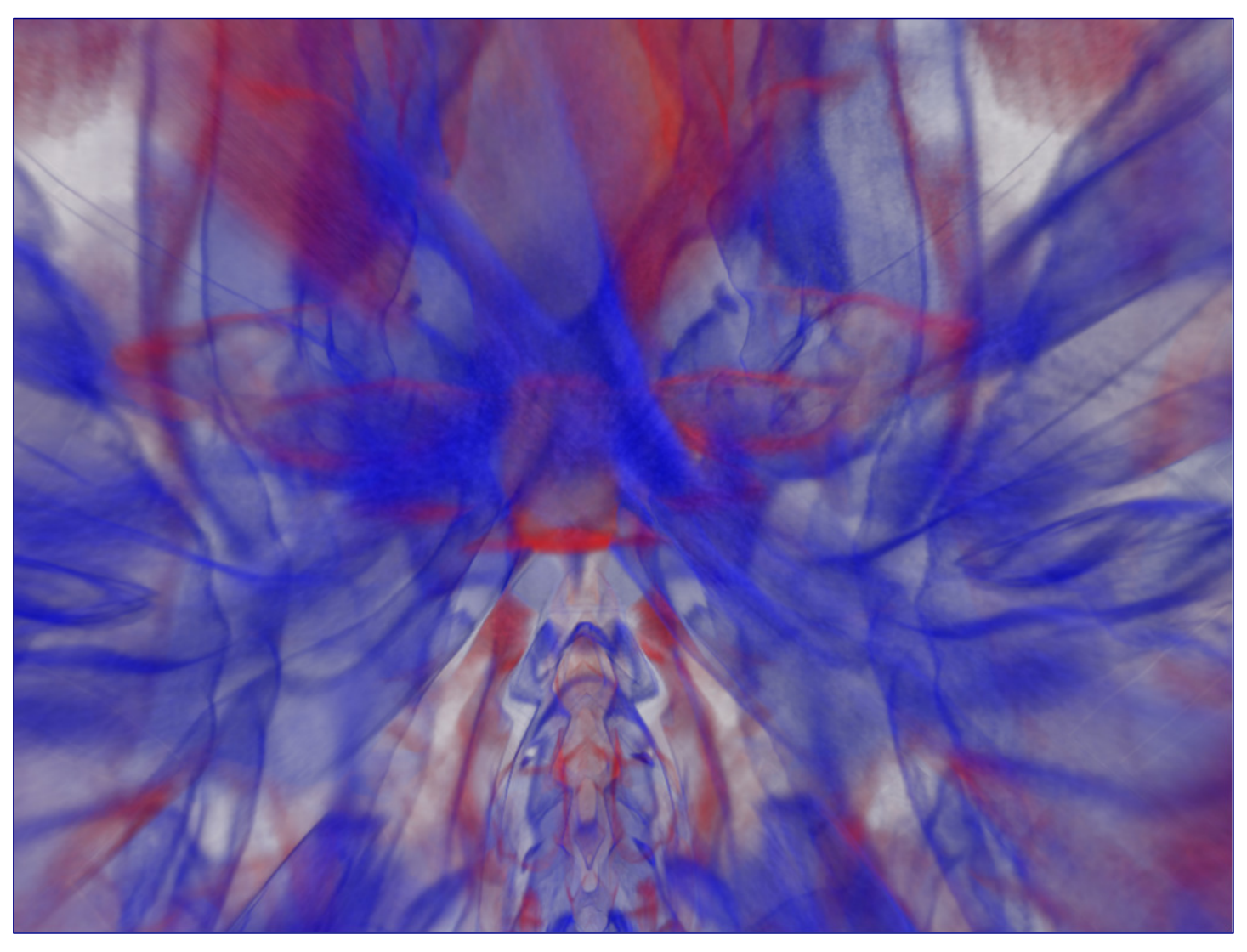


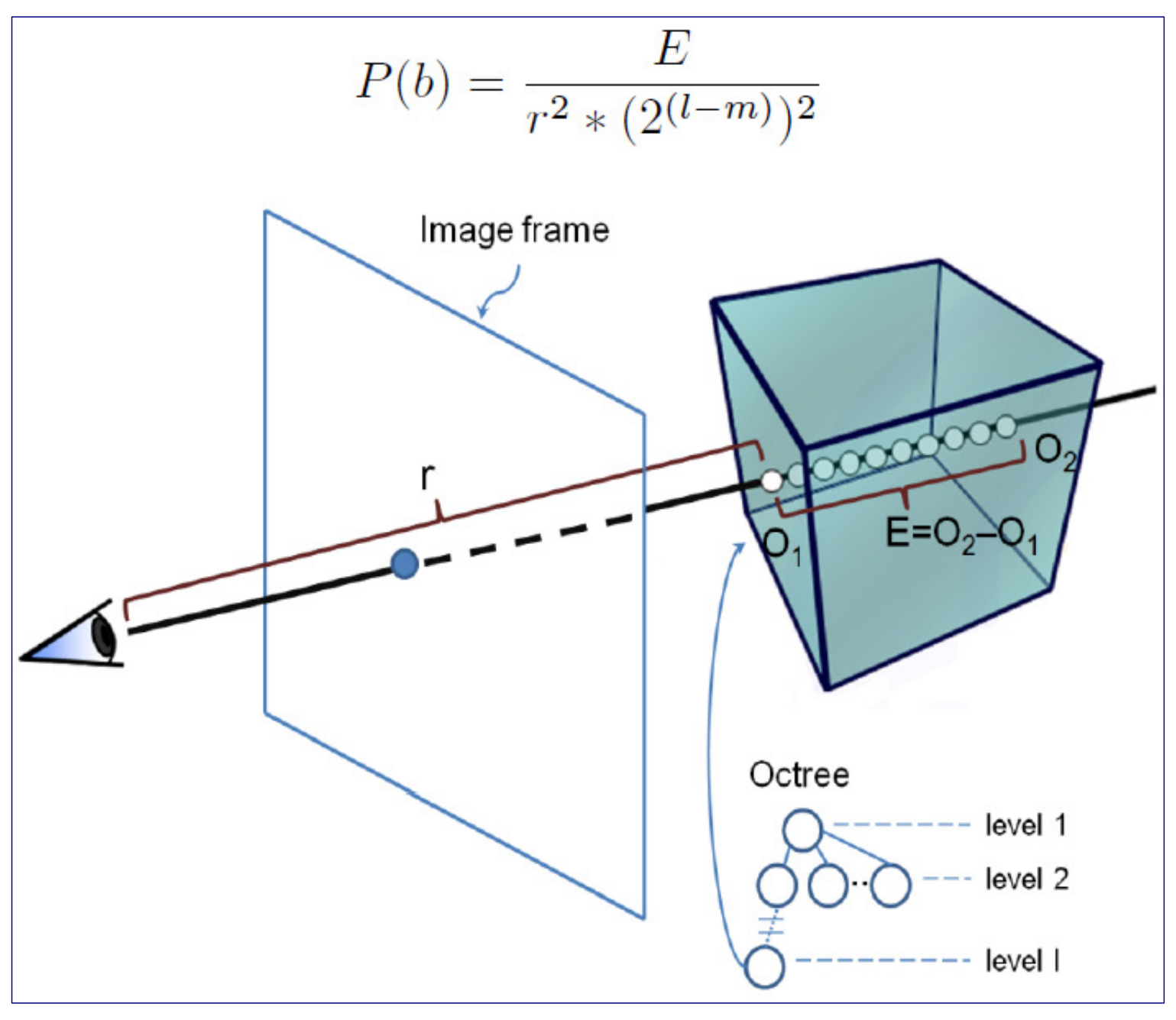






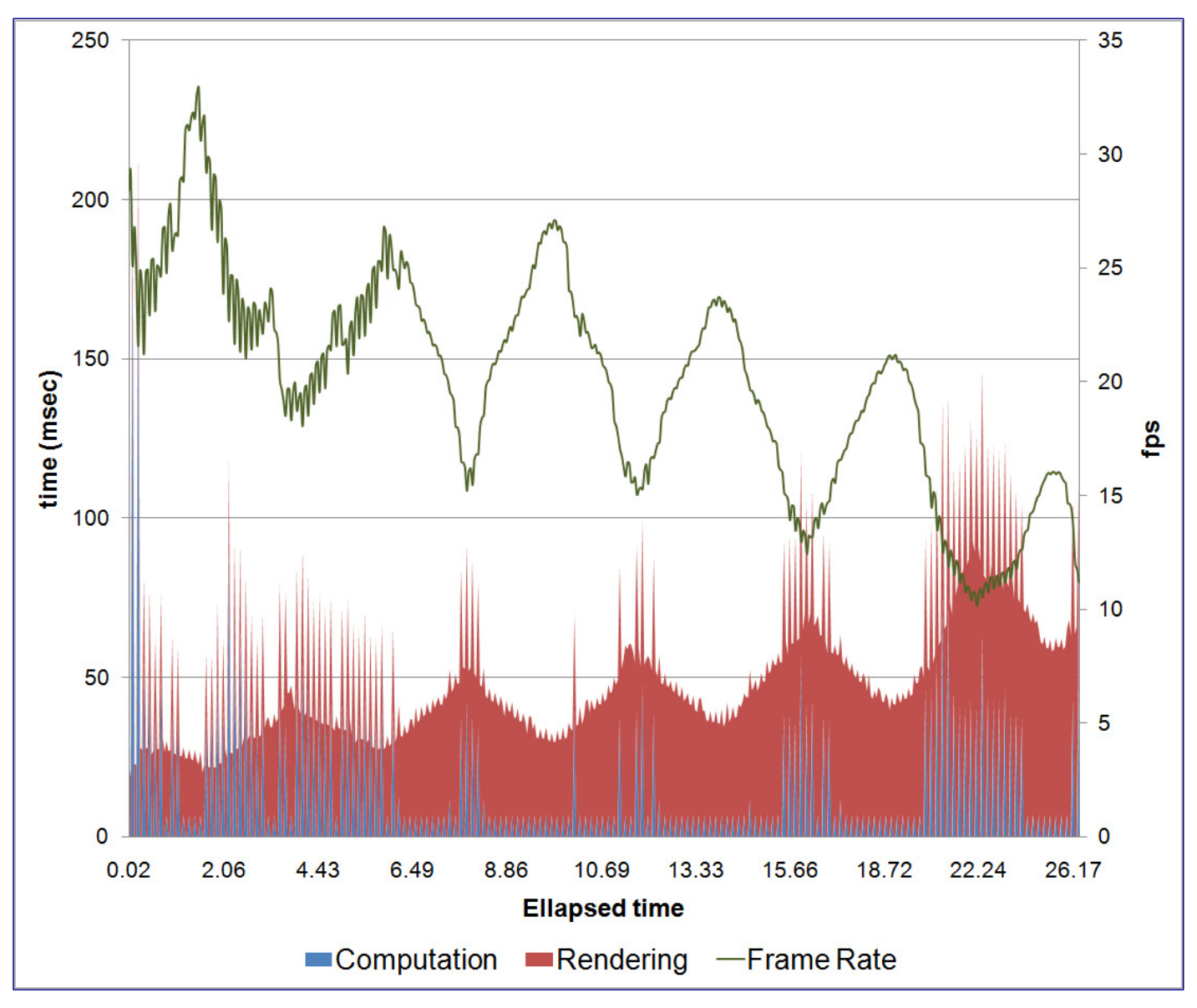






$$P(b) = \frac{E}{r^2 * (2^{(l-m)})^2}$$

We propose a novel data structure that extends the traditional texture octree and we introduce an optimality criteria based on the priority metric mentioned (See figure to the left). To meet the optimality conditions we use the *A* search algorithm* where we apply a set of operations in parallel on the GPU to the structures. Operations are performed based on a time and space budget over a sequence of frames in order to approach the target optimal tree. As the focus of the user is changing a new target tree is attempted. We evaluated the performance under different rendering conditions and for both steady and unsteady flows. Results indicate good frame rates unless heavy streaming is required. This method considerably outperforms any alternative technique and is the very first to turn LCS into an interactive, exploratory technique.

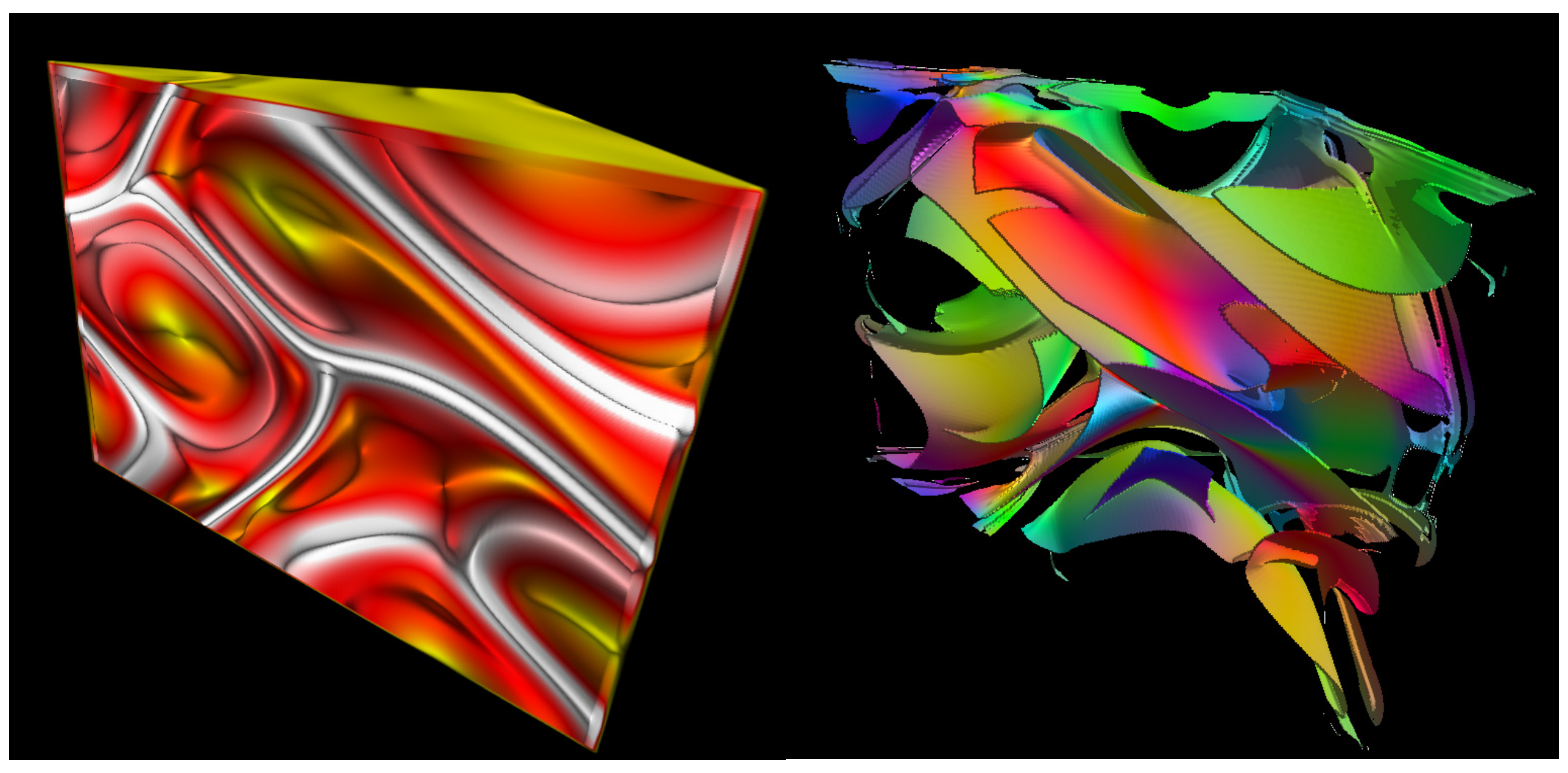


The effect of streaming on frame rates

	Jet 1	Jet 2	Delta wing
steady	20.03	18.96	12.77
unsteady + resident	18.93	15.86	13.2
unsteady + 2 x streaming	7.76	7.52	7.25
unsteady + 4 x streaming	4.95	4.95	5.01

The effect of viewport size on frame rates

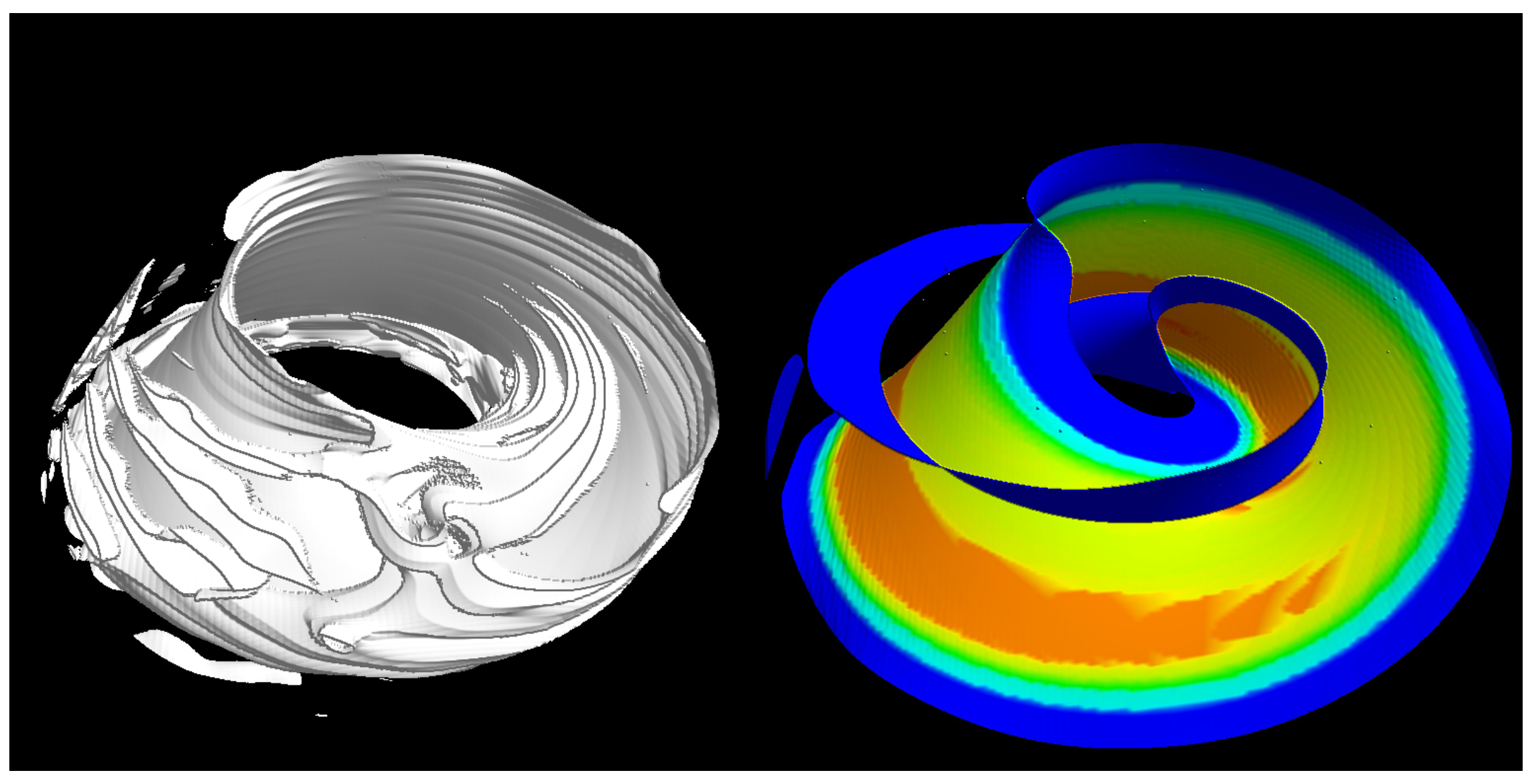
	Jet 1	Jet 2	Delta wing
400x300	48.62	43.01	39.37
640x480	28.83	25.79	19.63
1024x768	18.93	15.86	13.2

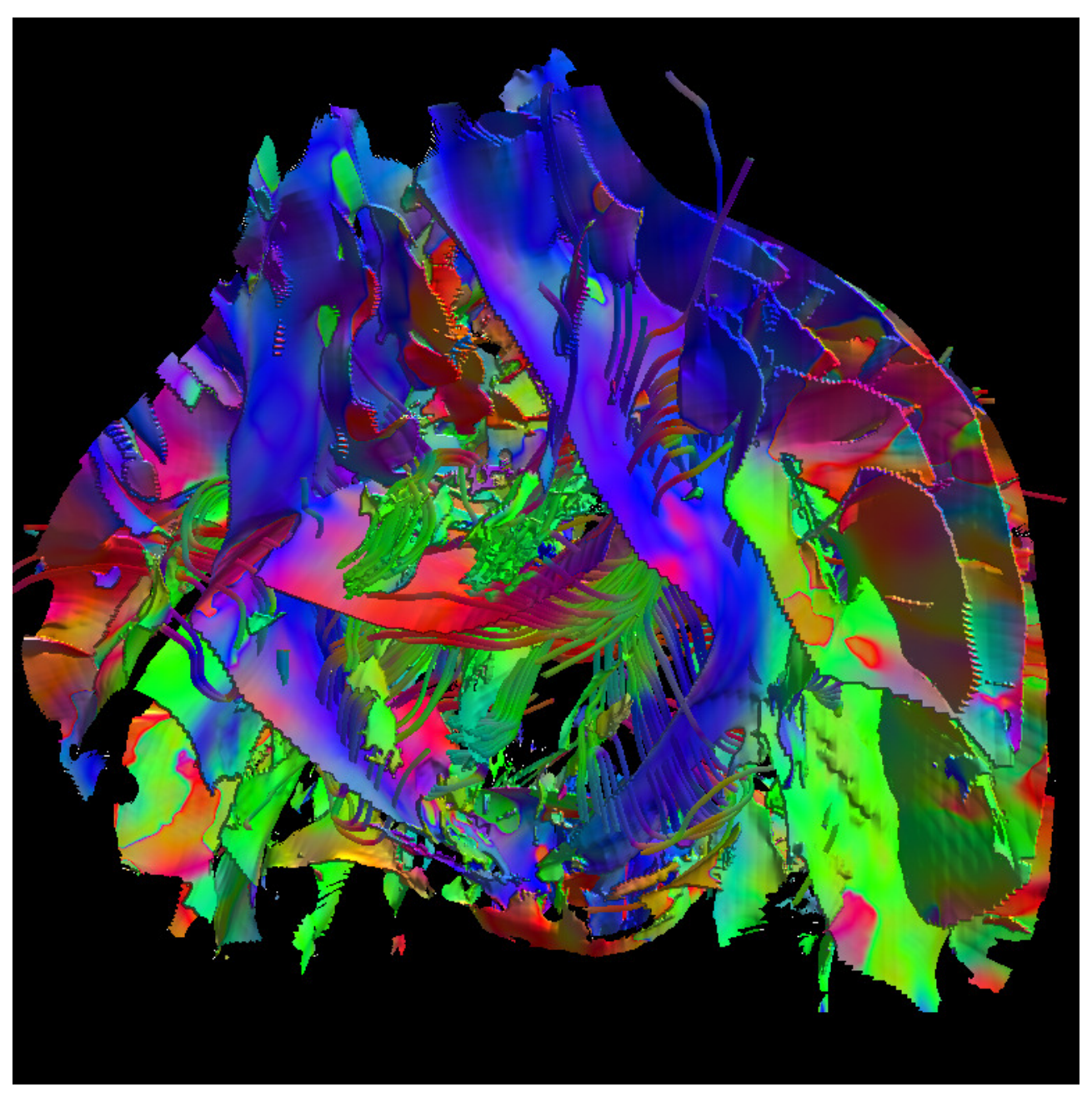


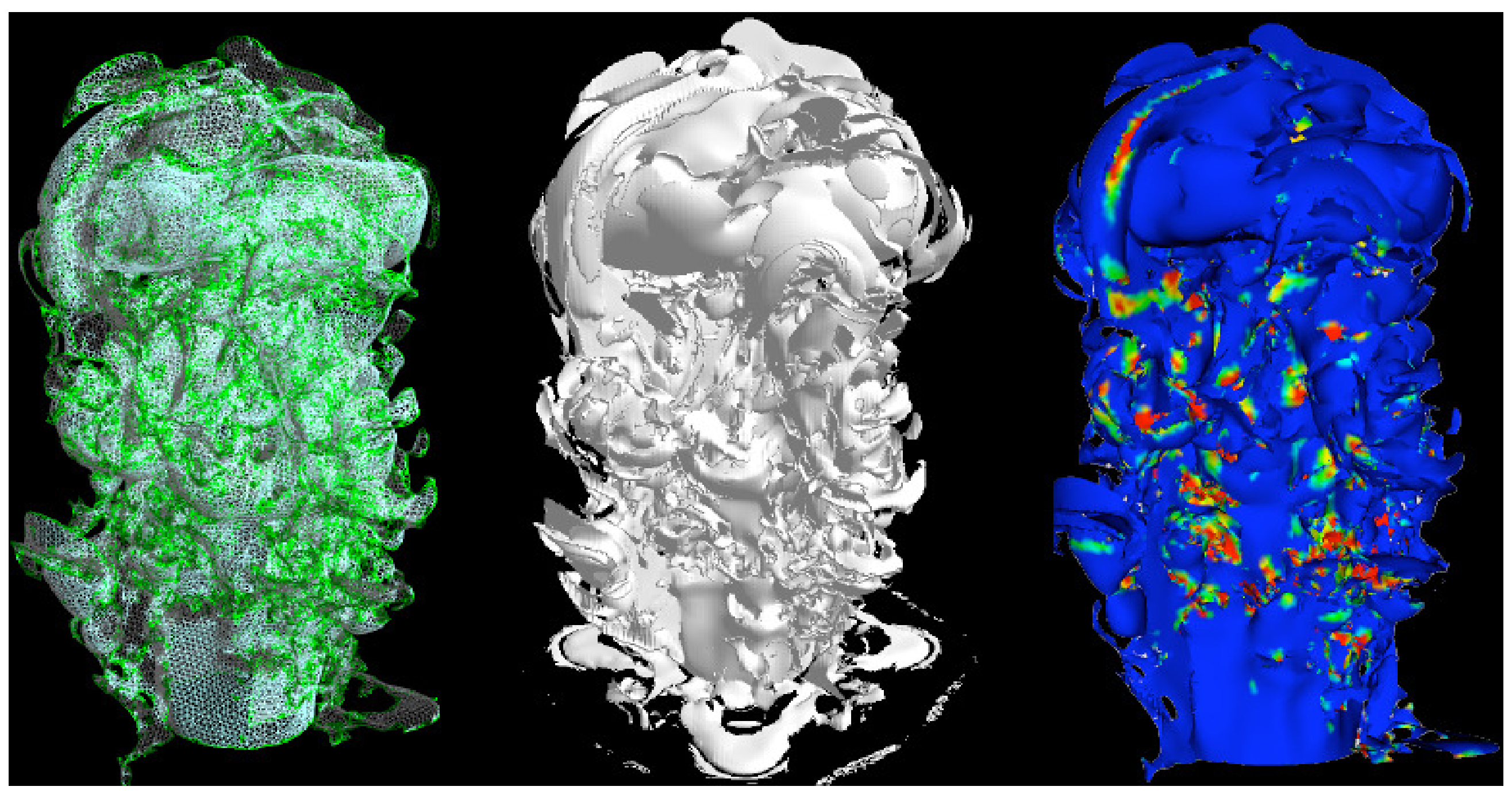
Efficient Visualization and Geometry Extraction of Crease Surfaces

Ridge and valley manifolds are receiving a growing attention in visualization research due to their ability to reveal the shapes of salient structures in numerical datasets across scientific, engineering, and medical applications. However, the methods proposed to date for their extraction in the visualization and image analysis literature are computationally expensive and typically applied in an offline setting. This setup does not properly support a user-driven exploration, which often requires control over various parameters tuned to filter false positives and spurious artifacts and highlight the most significant structures. Our method consists of a ray casting approach supporting a robust and efficient one-dimensional numerical search, and an image-based rendering strategy. Both efficiently implemented for the GPU to take advantage of the ray code path coherence, and texture caching and filtering on the GPU. The figure to the left shows the ridge surfaces for the ABC flow.

The search is performed along each ray by combining Newton's method with a backtracking strategy. The backtracking permits large steps along the ray while avoiding missing a surface intersection. Also, we use a two level skipping technique, implemented using textures, to reduce the computations. This is performed by pre-computing an underestimate of the crease strength across the volume and by creating a map indicating the distance to the closest surface from every voxel in the volume. Rendering is performed in a second phase by compositing the surfaces effects along rays. The normals are computed in screen space using least square fitting over a 3x3 neighborhood. By combining volume rendering at this phase, we produce an interactive enhanced visual representation of the data in which sharp salient manifolds are revealed in the fuzzy context of the embedding field. Figures to the right show the ray casting of the mobius band and the brain datasets in addition to the extracted geometry for the brain cancer dataset.







Using this ray-casting approach the user can tune the filtering parameters and select the structures of interest for which the geometry is extracted. A GPU-based advancing front algorithm is used for the meshing of crease surfaces. Excessive sampling and selection of mesh vertices is performed on the GPU to ensure accuracy and correct connections for a complex geometry without posing any memory overhead or complicate a meshing process on the CPU. See the figures to the left for the ray casting approach and the mesh extraction of ridges in the Jet flow.

$$\text{Ridge surface: } \mathbf{g} \cdot \mathbf{e}_3 = 0,\ \lambda_3 < 0$$

$$\text{Valley surface: } \mathbf{g} \cdot \mathbf{e}_1 = 0,\ \lambda_1 > 0$$