

Interactive 3D Visualizations of Laser Plasma Experiments on the Web and in VR

M. Danielová¹, P. Janečka¹, J. Grosz¹ and A. Holý¹

¹ELI Beamlines, Institute of Physics, Czech Academy of Sciences, Dolní Břežany, Czech Republic

Abstract

Lasers play essential role in many areas of both fundamental as well as applied research. Experiments in plasma physics that study complex interactions of intense laser beams with various forms of targets rely heavily not only on theory, but also on physics simulations. Such simulations, computed on a several thousand CPU core cluster, often trace the behaviour of billions of charged elementary particles over hundreds of time steps, generating up to terabytes of data. Visualization of the data is a challenging process that is mostly performed on GPU clusters using software such as ParaView [Aya15]. While this workflow has many advantages and provides advanced features for visualization and interpretation of the raw data, it is limited in the dissemination of the results on the web in interactive and VR mode.

Our approach at ELI Beamlines, a European laser research facility near Prague, uses our web-based interactive 3D visualization framework for exploration of curated and optimized simulation datasets. This application, while still in early stage of development, runs in a regular web browser and utilizes VR mode to offer scientists completely new point of view of their simulations. Based on in-house developed WebGL 2.0 implementation, the framework not only renders the dataset on the GPU in real-time at 60fps, but also provides orthogonal views, textual and numeric information, alongside a GUI containing timeline animation controls and layer visibility management, with additional graphical elements based on D3.js for plotting animated graphs and legends.

1. State of the Art

Examples of utilization of the power of GPUs for real-time 3D rendering inside web-browsers can be found across various fields. One example is the BioWeb3D application for interactive visualization of multiple layers of large biological datasets [PM13]. In physics, a browser-based 3D tool iSpy WebGL has been developed for displaying events at the CMS experiment at the Large Hadron Collider [McC17]. Aside from web applications focused on specific domains, visualization libraries provide solutions for rendering scientific data in web browsers using WebGL. We have evaluated Three.js javascript 3D library [thr] as well as VTK.js library [vtk] which has been developed directly for scientific visualization in browser. Unfortunately, both libraries could not render our datasets with hundreds of timeframes with sufficient performance. While Three.js would render only a small number of timeframes smoothly, VTK.js suffered from minutes-long loading times.

2. Visualization Workflow

In its current form, our visualization workflow based on data driven approach (Figure 1), allows us to share common code parts between each simulation visualization in terms of input data transformation and its rendering. The preprocessing step starts with raw data in .h5

format that are obtained from theoretical physicists responsible for simulation's lifecycle.

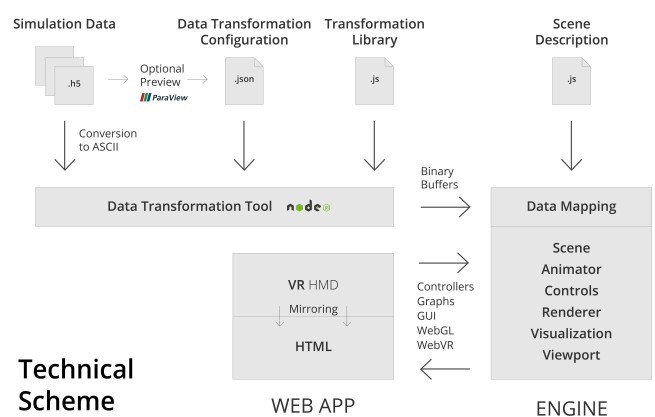


Figure 1: Detailed technical scheme of our visualization workflow

Input data are optionally previewed in ParaView software, to help us better understand their multidimensional characteristics and to determine which of their attributes are the most relevant for the visualization. Based on this or another a priori knowledge, data

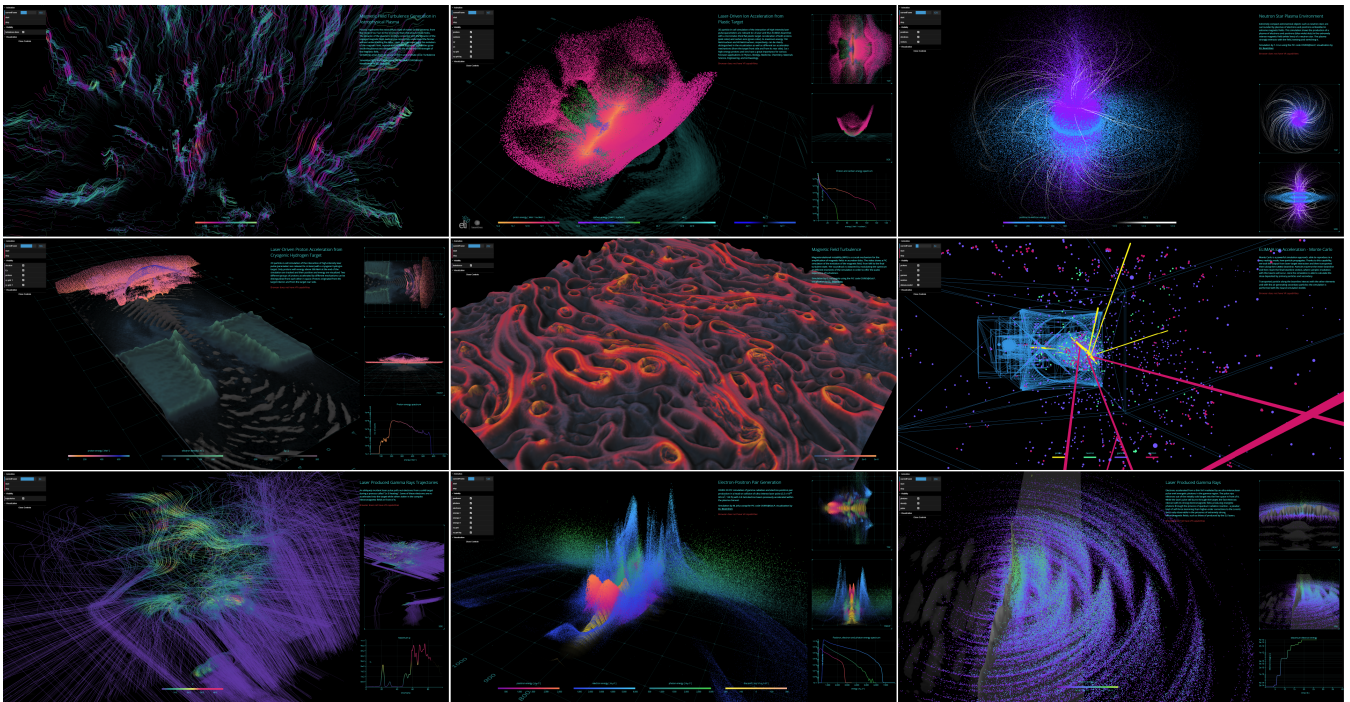


Figure 2: Screenshots of interactive web visualizations of various simulation datasets

transformation configuration is created. It describes which functions from transformation library and which parameters will be used by data transformation tool to generate binary buffers. Data transformation configuration is described in higher detail in the following subsection 2.1. The resulting buffers are stored on a web server acting as a data input for the visualization engine. Scene description, similarly as data transformation configuration, acts as a secondary input for the engine which is the core of visualization web app running as a client application inside regular web browser.

Additional layer containing information about the dataset as well as GUI in the resulting web page is formed by graphical elements. It utilizes d3.js library for graphs and legends and dat.GUI library for user interface that provide animation controls and layer visibility management. Examples of web applications with different simulation data are depicted in Figure 2.

Custom WebGL render solution drives visualization output to user's regular displays in a form of multiple viewports. Additionally, for users with access to VR headsets, WebVR technology is used to feed content to their HMDs. GUI controls together with VR controllers allow users to interact with displayed scene by filtering its content, moving and scaling it in 3D space, and navigating in animation timeline.

2.1. Data transformation configuration

Currently we have two main data types: particles and fields. The common setting for the particles is creation of XYZ position (in

case of a 2D dataset we map a specific simulated value such as energy to displacement), next we specify a color transfer function to calculate desired color, we also calculate weight of particles, encode it into alpha channel, and use it for particle sprite scale. If the original dataset is extremely large, we can also filter the particles to keep only certain portion of them. Lastly, if the particles have IDs through all the time frames we can calculate their trajectories and visualize them as lines.

2D field data can be handled and visualized in a number of ways: datasets can be cropped, downsampled, smoothed and colors assigned using color transfer function. In a similar way as for particles, pixels can be displaced to third dimension based on certain parameter we encode into the alpha channel of the texture. The result is a 3D surface. To enhance its spatial features we calculate ambient occlusion value by which we modify each pixel's color. Our vision is to enable visualization of 3D volumetric data in the near future.

3. Conclusion and future work

Scientists who are carrying out both 3D and 2D simulations of laser plasma experiments appreciate the ability to explore their datasets interactively, especially in the VR mode that offers them completely new perspective of physics processes they study. Future development is focused on providing users a web-based toolkit for setting up and configuring their visualization themselves, implementation of additional visualization and UI features as well as compression solutions for efficient transfer of data over the internet.

References

- [Aya15] AYACHIT U.: *The paraview guide: a parallel visualization application*. Kitware, Inc., 2015. 1
- [McC17] MCCAULEY T.: Iop: A browser-based event display for the cms experiment at the lhc using webgl. In *J. Phys.: Conf. Ser.* (2017), vol. 898, p. 072030. 1
- [PM13] PETTIT J.-B., MARIONI J. C.: bioweb3d: an online webgl 3d data visualisation tool. *BMC bioinformatics* 14, 1 (2013), 185. 1
- [thr] Three.js. <https://threejs.org/>. Accessed: 2019-05-06. 1
- [vtk] Vtk.js. <https://kitware.github.io/vtk-js/docs/>. Accessed: 2019-05-06. 1