

Three Novel Software Tools for ASDEX Upgrade

S. Martinov^{a,1}, T. Löbhard^{b,2}, T. Lunt^{a,1,3}, K. Behler^{a,†}, R. Drube^a, H. Eixenberger^a,
A. Herrmann^a, A. Lohs^a, K. Lüddecke^c, R. Merkel^a, G. Neu^a,
ASDEX Upgrade Team^a, MPCDF Garching^d

^aMax-Planck-Institut für Plasmaphysik, Boltzmannstr. 2, D-85748 Garching bei München

^bwork carried out in 2012 as affiliate to the Max-Planck-Institut für Plasmaphysik

^cUnlimited Computer Systems GmbH, Seeshauptstr. 15, D-82393 Iffeldorf

^dMax Planck Computing and Data Facility, Boltzmannstr. 2, D-85748 Garching

¹equally contributing authors of AUGtv, ²main author of SIOSzi, ³main author of AUGpy, [†]corresponding author

Visualization of measurements together with experimental settings is a general subject in experiments analysis. The complex engineering design, 3D geometry, and manifold of diagnostics in larger fusion research experiments justify the development of special analysis and visualization programs. Novel ASDEX Upgrade (AUG) software tools bring together virtual navigation through 3D device models and advanced play-back and interpretation of video streams from plasma discharges. A third little tool allows the web-based platform independent observation of real-time diagnostic signals. While all three tools stem from spontaneous development ideas and are not considered mission critical for the operation of a fusion device, they with time and growing completeness shaped up as valuable helpers to visualize acquired data in fusion research. A short overview on the goals, the features, and the design as well as the operation of these tools is given in this paper.

Keywords: 3D model viewer, scientific video display, data visualization, web-based data oscilloscope, mobile data display, public domain python tools

1 Introduction

In 25 years of operation ASDEX Upgrade's (AUG's) data acquisition has grown by a factor of 10000 from megabytes to tens of gigabytes. So respectively the number and kinds of diagnostics, the number of channels per diagnostic, and the temporal resolution per channel have grown. During this time the experiment has undergone regular redesigns in many aspects concerning e.g. the geometry of the divertor, the interior wall details, the location and set-up of diagnostics, and last not least the kind of diagnostics. Latter today comprise a reasonable number of video diagnostics in the infrared and visible spectral range. These kinds of “novel” diagnostics were not supported by the original data analysis and visualization tools. While some generic viewers exist for these kinds of data streams, with an increasing number of streaming channels one soon finds out, the generic tools do not suffice for a scientific view to the data.

These various modifications and renovations of the experiment over time led to a couple of developments on the data analysis side of which the here presented three software solutions because of their originality and universal applicability seem of particular interest for a broader fusion science community. They are mainly based on publicly shared software developments providing great benefit for this kind of scientific applications.

2 AUGpy - Mapping Diagnostics Views into 3D Construction Design

AUGpy is a 3-dimensional visualization application developed as a follow up application to AUGddd [1] now written in Python. It combines virtual CAD model data of the ASDEX Upgrade Tokamak plasma vessel and built-in components together with other geometry data from machine design and diagnostics. AUGpy allows to virtually walk through this combined spatial data model on screen. Vessel and diagnostic structures as well as coordinate lines, lines of sight, viewing planes of diagnostics and cameras are superimposed at the right positions. Even experimental data defined on spatial coordinates like profiles or images may be shown hovering in the model at the corresponding locations.

2.1 How AUGpy works

AUGpy basically is a browser engine for 3D virtual models stemming from mechanical CAD construction works. For visualizing a full Tokamak all the model files describing the mechanical structure are to be provided as StereoLithography (STL [2]) files. At start-up the user can choose at which detail level (full, medium, or small) the visualization shall take place and which parts of the model tree construction layers shall be visible or hidden. It is also possible to choose colors for highlighting selective layers

[†] corresponding author email: karl.behler@ipp.mpg.de

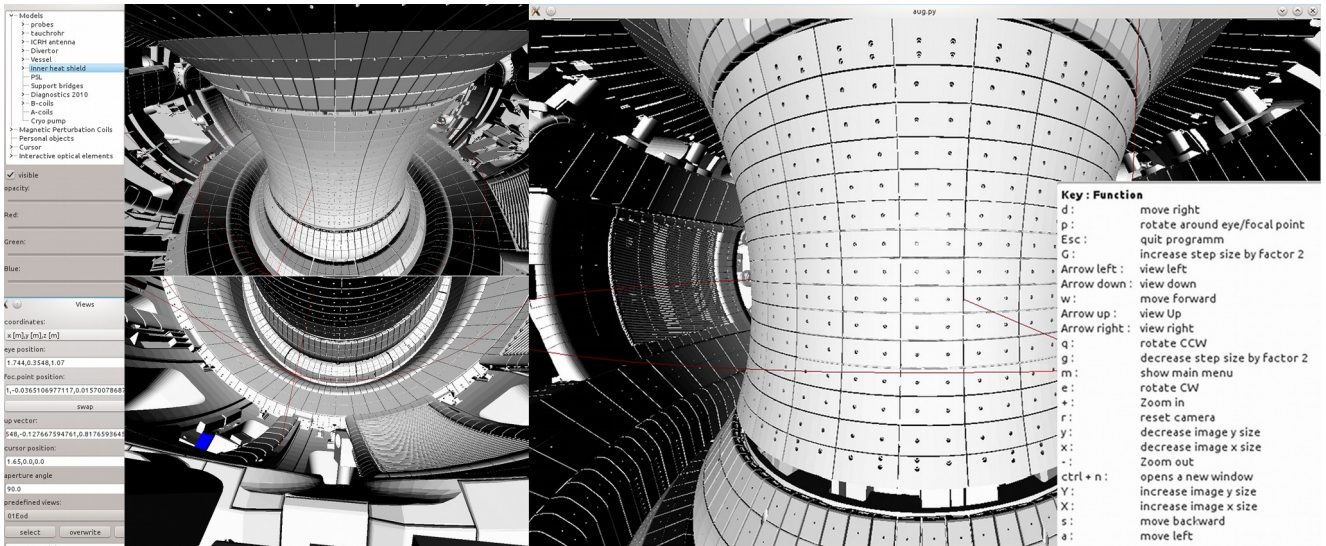


Fig. 1: Virtual 3D navigation inside the ASDEX Upgrade vessel model. Three different perspectives of the inner heat shield and looking down to the divertor are shown. The menu insets left and right give a glance about the user control possibilities of the tool.

of the model. Three Tokamak models are already available for AUGpy: AUG, JET, and TCV.

Navigation in three dimensions is very similar to that in CAD programs but requires no special 3D human interface devices. Hereby one may manually navigate reading coordinates from the indicator until a certain view is reached or one may input a set of coordinates as numbers to choose eye, focus point, and viewing angle. For the coordinates reading one may choose from three systems. (Figure 1 gives an example of the AUGpy user interface.)

Views defined in this way may be saved by the user under names representing his favorite constellation for

various purposes e.g. representing a diagnostics view. The program comes with a set of predefined views representing the various perspectives of the video cameras installed inside the experiment (see section about AUGtv).

The transparent mapping of 2D images into the 3D space of the experiment model allows the overlay of diagnostics or camera pictures over the visible Tokamak structures. By adapting the 3D viewing parameters the exact alignment between a given 2D image and the vessel structure is possible (Figure 2). Knowing the exact camera position this allows to identify the exact viewing angle and rotational orientation and thus the spatial calibration of the 2D images a camera takes. In fact with this tool the exact line of sight in vessel coordinates for each pixel of a

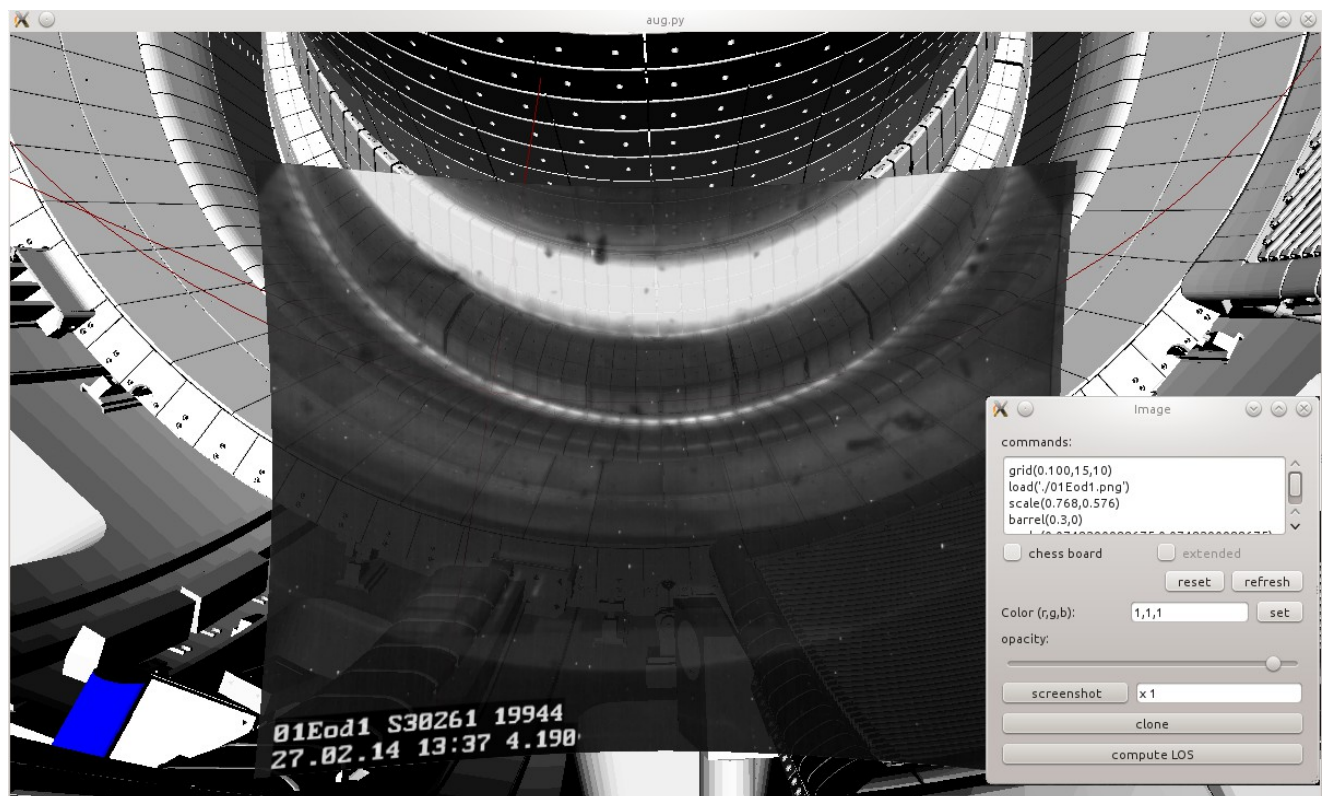


Fig. 2: Mapping a camera image into the 3D model applying geometrical adaptations. (see text)

video or diagnostic image is determined.

The program also features the generation of wireframe and 2D projection images of the vessel structure to be used as overlays (backgrounds) for viewing images taken in the darkness of the vessel or in spectral ranges where the vessel structures normally are not visible. This helps the human eye to interpret images and videos taken under this conditions. This kind of overlays are used in AUGtv to support the understanding of the ASDEX Upgrade videos.

2.2 Software design details of AUGpy

AUGpy is completely written in Python. It makes use of many publicly available function packages such as pyvtk, PyQt, numpy, matplotlib, opencv, and others. Being widely platform independent by design AUGpy has been tested on Windows, MacOS, and Linux. Using the OpenGL standard for rendering enables 3D graphics acceleration not only on local hardware but also remotely via Virtual GL [3].

3 AUGtv - A Scientific Multi-Video Viewer

AUGtv is not just another video viewer. Combining a sophisticated user interface, an optimized file caching and memory management, various real-time scientific filtering algorithms, the ability to decode proprietary science video formats, and a bunch of diagnostics metadata for each video stream it implements a visualization tool highly tailored for scientific exploration in fusion research. Especially the immediate quantitative analysis features of AUGtv in combination with the spatial, temporal, spectral and intensity calibration of the camera channels are making video inspection highly relevant for the immediate understanding of physics.

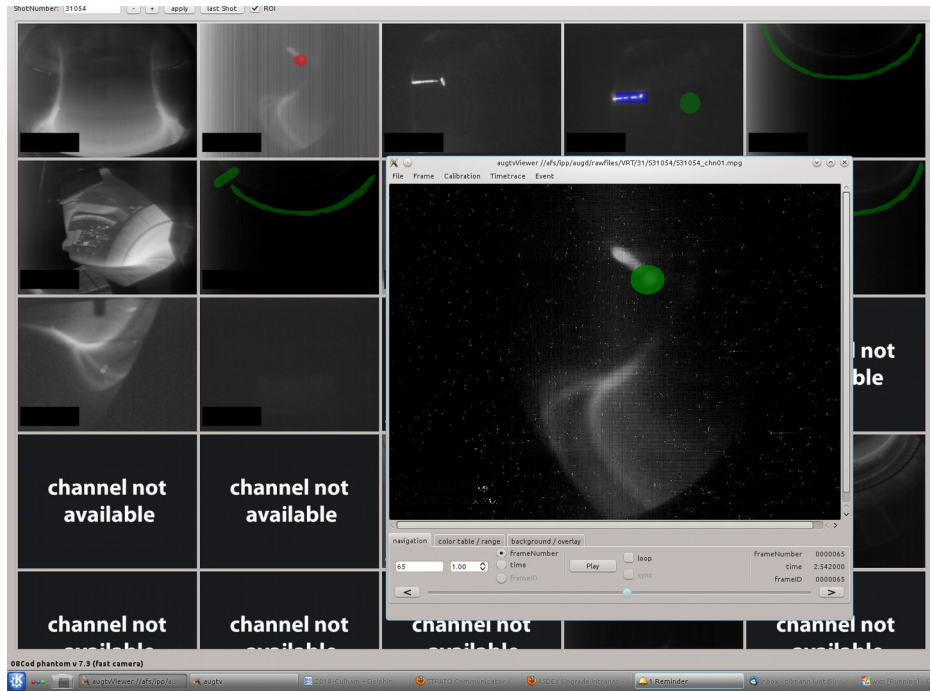


Fig. 3: AUGtv tiled overview and video window in foreground.

3.1 AUGtv workflow

Entering a plasma discharge number AUGtv starts with a tiled still image overview (Fig. 3) of the videos available while the status line gives some first information about the video channel the cursor is hovering over. Using an optimized averaging algorithm for creating the still images presented on this first screen, it already provides a lot of information on how a plasma discharge was run and which videos might be of particular interest. Also these still images depict the regions of interest (ROI) defined in the surveillance systems. These ROI are shown as colored areas indicating the surveillance mode and state as follows: green areas are actively checked for intensity peak, but have not reached the surveillance threshold during the discharge. Red areas indicate an ROI which has triggered a premature end of discharge because of an over-the-threshold radiation intensity. And blue areas are currently inactive. Thus already the first screen allows a rough quantitative assessment of a plasma discharge and an easier decision which video to view in details. Also accessible from the tiles is more meta data information as shown in Fig. 4.

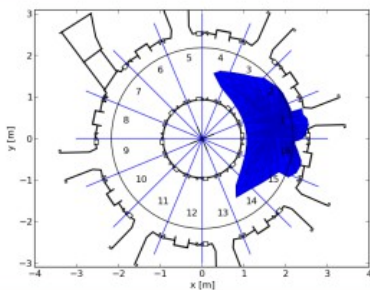
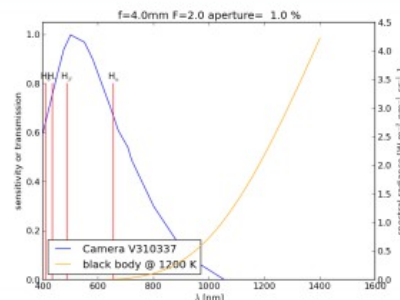


Fig. 4: From the tiled overview also more meta data about the video channels is easily accessible. Graphs depicting the camera perspective, diagrams showing the spectral filtering, and images of the vessel background can be shown.



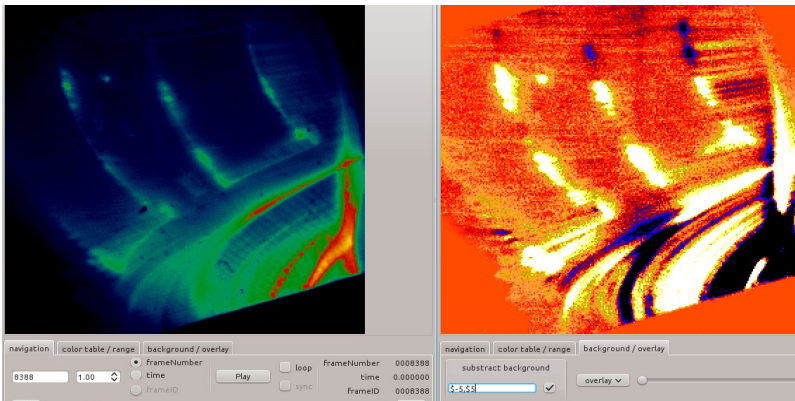


Fig. 5: Two different false color mappings of the same video to allow simultaneous in depth inspection of two small intensity ranges.(To view this image in full color please refer to the online version of this paper.)

Now clicking on one of these images opens a new window for the selected stream allowing not only playing but real-time modifying the video for optimal visualization. To achieve this couple of transformations can be applied in real-time to enhance the visibility of processes or events going on in the plasma: background

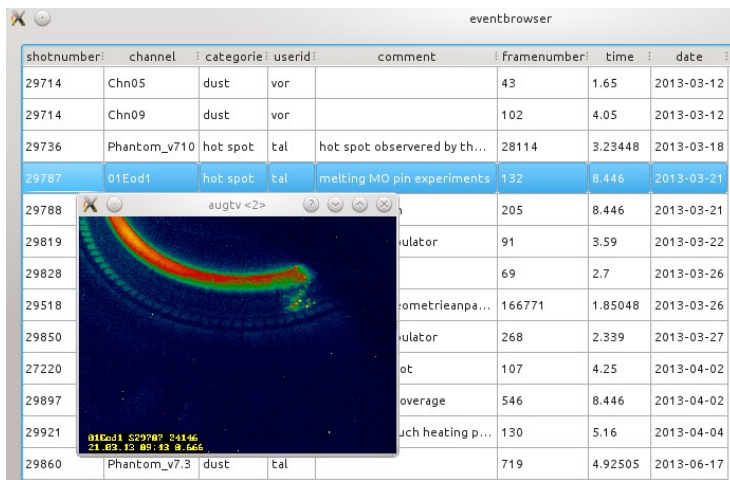


Fig. 6: Using the event browser to find image sequences of particular interest in the huge heap of videos.

subtraction based on single or floating average frames, rescaling of contrast and luminosity, mapping of the dynamic range to false colors, and overlays indicating ROI or wireframe structures from the CAD plasma vessel model (see AUGpy).

For most streams the absolute time of the frames may be loaded from meta data storage. The "time" button together with the "sync" flag allows to play multiple streams synchronously in parallel. Frame step and frame rate can be chosen in a wide range from slow motion to rapid scan. Also looping through a defined frame range or single stepping is possible. For enhanced visualization even the same video can be played in two different windows applying different transformations for

emphasizing two different aspects in the same video in parallel (Fig. 5).

Proprietary scientific video formats thus as used for fast imaging or infrared thermography measurements have been implemented as a first goal. But also consumer video formats like mpeg, mjpg, and others are supported to allow the usage of standard cameras.

To provide a better orientation in the large video data collection and especially when viewing data from fast imaging cameras delivering 50000 frames from a 10 seconds plasma discharge an event database allows the annotation of video streams by authorized users and helps others later to navigate searching for plasma discharges with events relevant to particular research. Figure 6 shows the event browser interface.

3.2 Software design details of AUGtv

The GUI of the application is written in QT for versatility and portability. Video decoding and algorithms are realized using public packages such as avcodec, libmpeg, opencv, Python, and C++. The latter especially is used where performance comes into play for effective scientific decision making. Currently AUGtv is compatible with Windows, MacOS, Linux, and Solaris. Future support of hand held devices seems possible.

4 SIOszi - A Web Based Real-Time Diagnostics Oscilloscope

SIO-Oscilloscope (SIOszi) provides immediate data visualization for Serial I/O (SIO) based data acquisition (DAQ) systems. The SIO DAQ system has been described earlier [4]. Its main purpose is to deliver data from measuring periphery such as ADCs directly into the main memory of a diagnostics computer from where it is eventually processed in real-time and finally stored into the AUG shot files archive. However, to visualize data while setting-up measuring channels in laboratory an application smaller and simpler

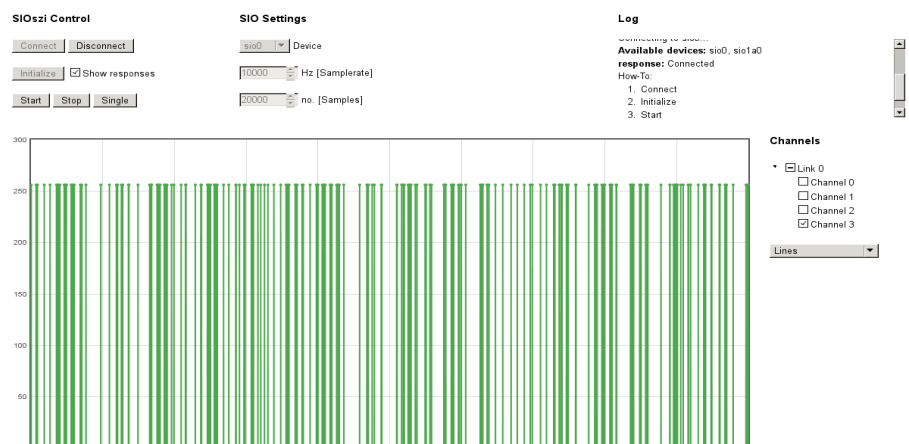


Figure 7: An actual browser screen shot from SIOszi showing a 10 kHz digital signal oscillating between two values (0 and 255) over two seconds. This is available on hand held devices as well.

than a full diagnostic is required. Providing kind of an oscilloscope functionality just the ability to decode the DAQ data stream in real-time and to extract and visualize particular signals is required. SIOszi is such a stand-alone application written in Python and allowing to use a diagnostic like a specialized oscilloscope. An example browser screen shot is given in figure 8. Since only an HTML browser is required for visualization even a smartphone may be used as display to operate SIOszi.

4.1 SIOszi operation

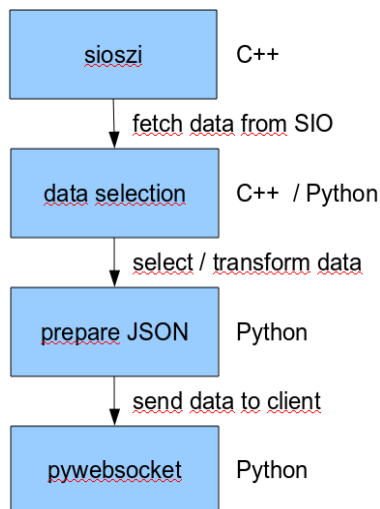


Fig. 8: A schematic view of the SIOszi data flow.

The SIOszi kernel routines directly talk to the measuring devices using the SIO application interface library. This functional kernel is embedded into a small web-socket server providing web functionality and HTML5 graphs to any kind of web client. All what is needed is an HTML5 enabled browser. So SIOszi allows the immediate inspection

of measuring channels for the adjustment and testing from anywhere on the network to virtually any web enabled device.

Acknowledgment

This work has been carried out within the framework of the EUROfusion Consortium and has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement number 633053. The views and opinions expressed herein do not necessarily reflect those of the European Commission.

Future plans are to read the diagnostics settings and channel names from the configuration file of the respective diagnostic to allow addressing data by signal names and calibrating signals to physical units. The current presentation of signals as simple graphs may be extended in future by diagnostic specific user contributed Python analysis routines.

4.2 Software design details of SIOszi

The SIOszi server is written in C++ and Python using the pywebsocket package [5]. It delivers HTML5 and java-script [6] code to web browsers to initialize the communication between client and server as well as to update the graphical representation periodically using a web socket connection. The server requires access to the respective ASDEX Upgrade DAQ hardware and SIO system libraries. The client just needs a modern web browser. The basic program data flow is given in figure 8.

5 Summary and Outlook

Innovative software tools are provided to the ASDEX Upgrade fusion research community thanks to publicly available sophisticated program library packages. However, special effort was made collecting and implementing features of particular relevance for the particular research tasks.

Remote visualization and web techniques enable the application of today's tools on a variety of platforms partially including hand held devices. GIT repository access can be provided on request for readers interested in using or contributing to the development of these applications.

References

- [1] T. Lunt et al., <http://dx.doi.org/10.1016/j.nima.2010.04.150>
- [2] StereoLithography Interface Specification, 3D Systems, Inc., July 1988
- [3] VirtualGL (VGL), <http://www.virtualgl.org/>
- [4] K. Behler et al., <http://dx.doi.org/10.1016/j.fusengdes.2012.09.020>
- [5] pywebsocket, <https://code.google.com/p/pywebsocket/>
- [6] HTML5 plot library, <http://www.flotcharts.org/>