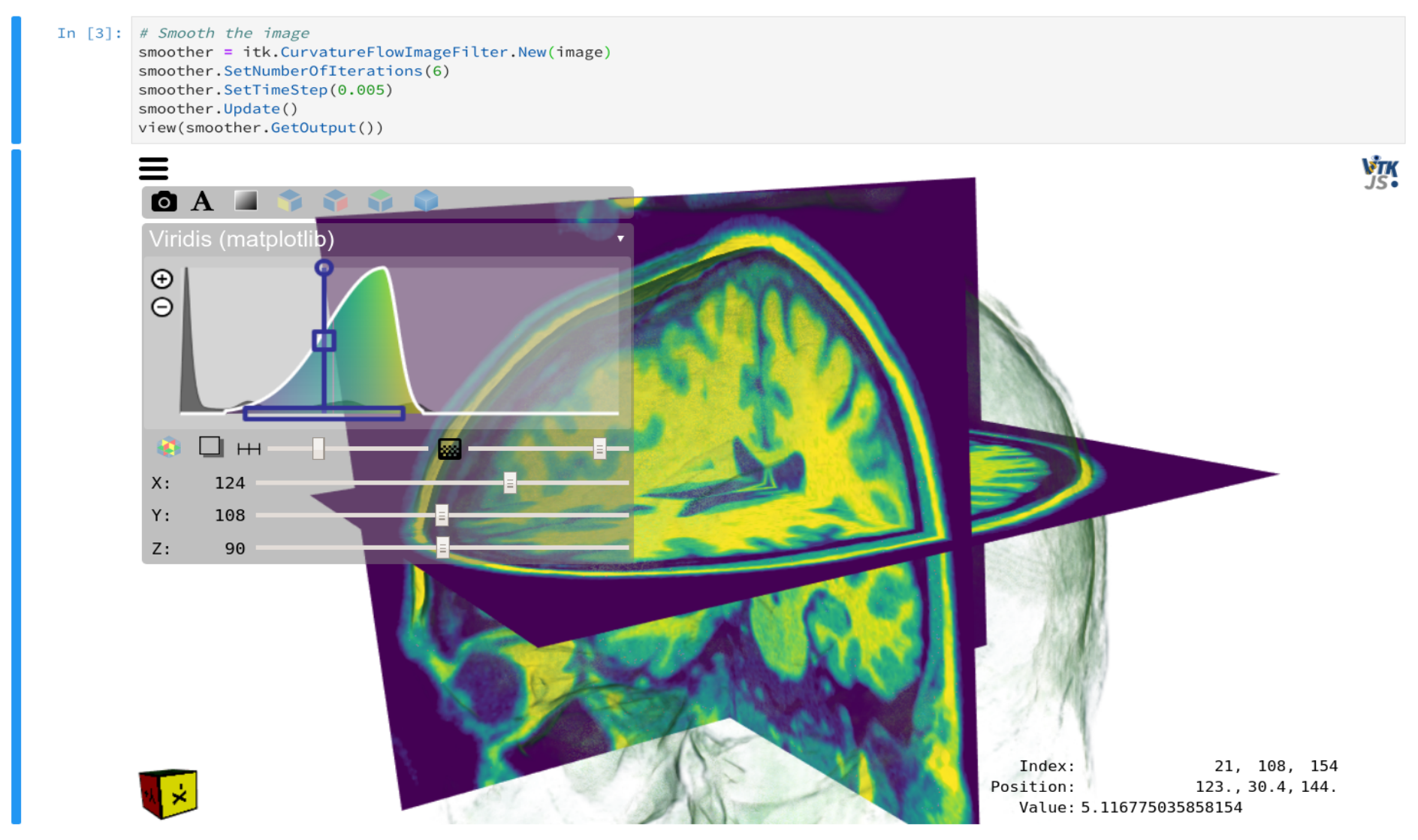
itk-jupyter-widgets: Interactive 2D and 3D Image Visualization for Jupyter

Matthew McCormick[1*], Sebastien Jourdain[1], Ken Martin[1], Alexis Girault[1],
Michael Grauer[1], Jean-Christophe Fillion-Robin[1]
1 *Kitware, Inc, USA;*
* matt.mccormick@kitware.com

Introduction

We present a new package for interactive visualization and inspection of 2D and 3D imaging-related datasets in the Jupyter Notebook and JupyterLab: ***itkwidgets***.

Leveraging the capabilities of WebGL, these widgets offer browser-based inspection of imaging data for researchers working in the medical, biological, materials science, geoscience, and other scientific domains.



```
In [3]: # Smooth the image
        smoother = itk.CurvatureFlowImageFilter.New(image)
        smoother.SetNumberOfIterations(6)
        smoother.SetTimeStep(0.005)
        smoother.Update()
        view(smoother.GetOutput())
```

```
Index:       21,  108,  154
Position:    123., 30.4, 144.
Value: 5.116775035858154
```

Human brain volume rendering and slices. Upper left: user interface controls for window/level, opacity transfer function, slice position, etc. Lower right: data probe image index, position, and value.

Methods

These widgets are built on *itk.js* and *vtk.js*, JavaScript and WebAssembly interfaces and implementations of the Insight Toolkit (ITK) and the Visualization Toolkit (VTK).

The widgets are designed to support image analysis with ITK, but they work equally well with other spatial analysis tools in the scientific Python ecosystem, such as scipy.ndimage, scikit-image, ImageJ-ImgLib2, Dask, OpenCV, or VTK, as a result of ITK's NumPy interoperability layer.

DASK
OpenCV
ITK
VTK

The package is open source, licensed with the Apache 2.0 license, and developed with the SciPy community.

Videos: https://goo.gl/7wgpzh

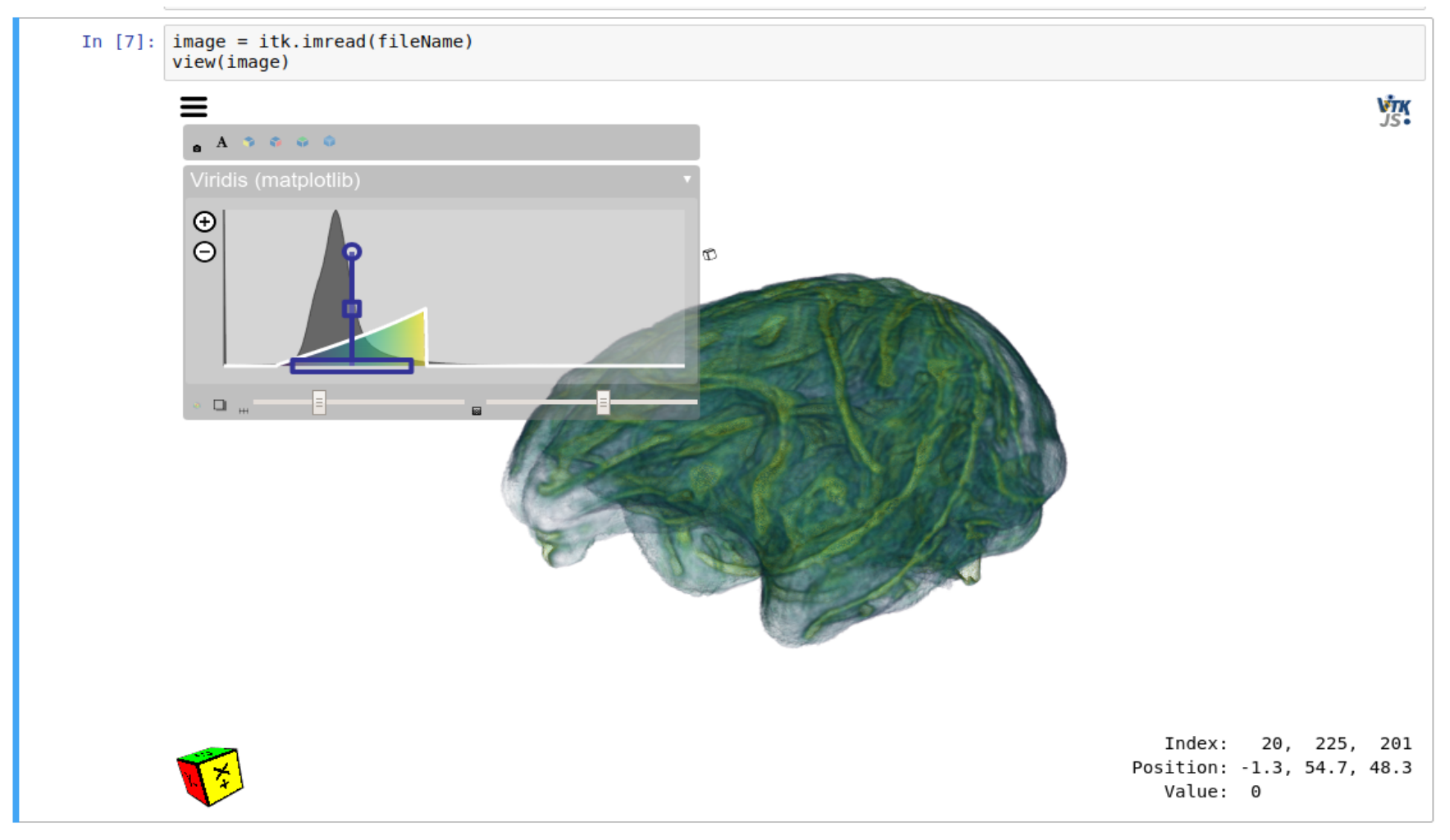
Results

The widgets are capable of visualizing large 2D and 3D images with all common plain old datatype (POD) pixels types, anisotropic spacing, and non-axis aligned orientations.

Customizable volume rendering is available, including adjustable gradient-based opacity settings. Window, level, and complex opacity transfer functions are quickly selected with a piecewise-Gaussian transfer function editor.

Dozens of colormaps are available through the widget's graphical interface to enable different perspectives on the data. For 3D data, triplane and individual image slice views enable direct inspection of the data along with a cursor to identify the local pixel index, position, and value.

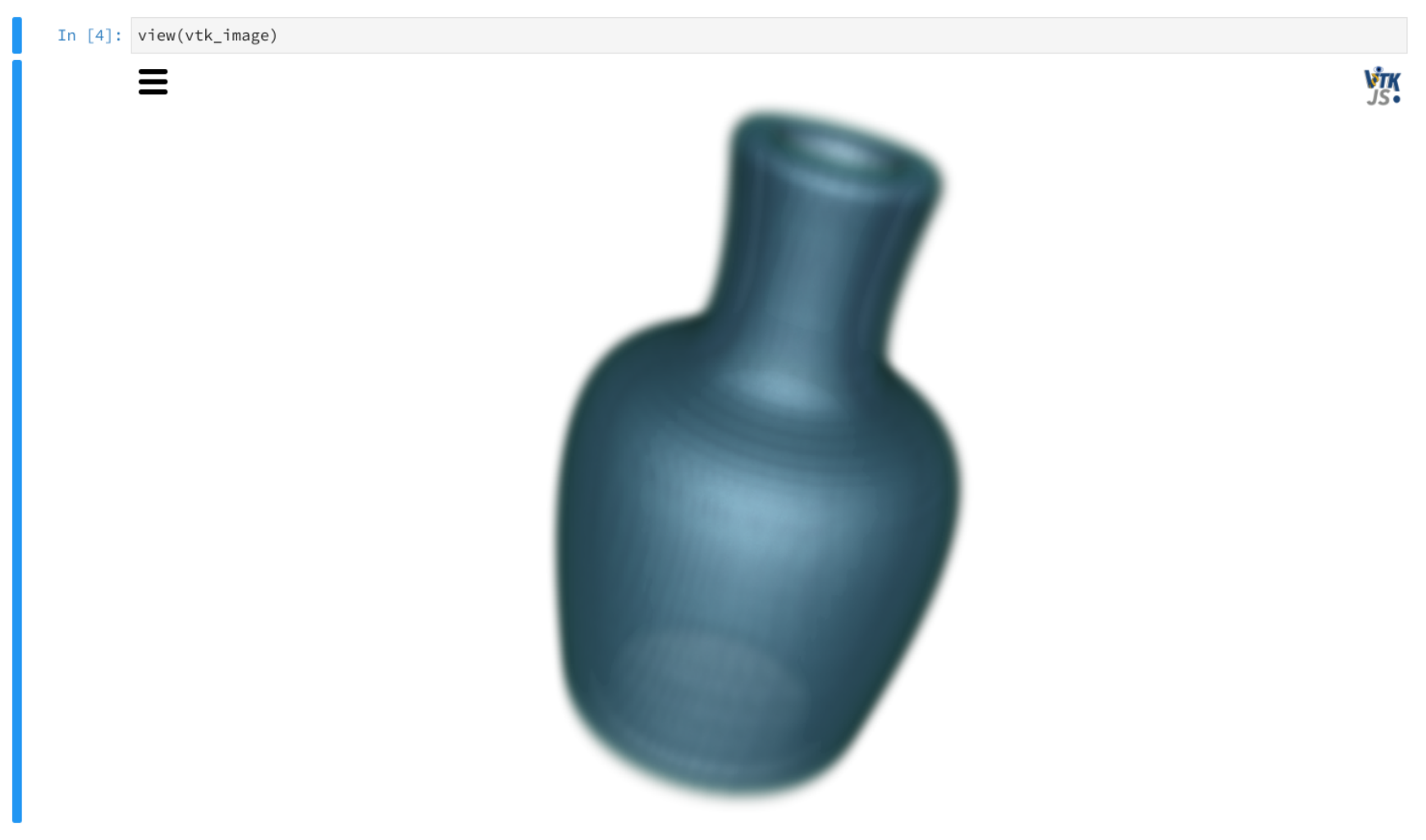
Server-to-client data transfer delays were minimized via fast pixel buffer compression and decompression with binary Python and WebAssembly interfaces to the *zstandard* compression library.



```
In [7]: image = itk.imread(fileName)
        view(image)
```

```
Index:     20,  225,  201
Position: -1.3, 54.7, 48.3
Value:    0
```

Monkey brain volume rendering.
Original image size: 52 MB
*Compressed size: **8 MB***
Decompression time: 193 ms



```
In [4]: view(vtk_image)
```

Vase volume rendering.
Original image size: 871 KB
*Compressed size: **57 KB***
Decompression time: 87 ms

Discussion and Future Work

By leveraging WebGL and WebAssembly, *itkwidgets* provides high-performance, interactive visualization of scientific datasets in the web browser. Jupyter's tight coupling of the myriad of powerful, open-source image analysis libraries available in Python with this visualization tool facilitates a rapid understanding of imaging datasets and the impact of processing operations.

Network bandwidth and latency along with an increasing size of scientific imaging data presents new challenges. Fast compression can help address these issues.

Future work includes support for meshes and point sets, dynamic decimation of large datasets, and visualization state serialization.

```
pip install itkwidgets
```

https://github.com/InsightSoftwareConsortium/itk-jupyter-widgets