The Journal of Open Source Software

PyVista: 3D plotting and mesh analysis through a streamlined interface for the Visualization Toolkit (VTK)

C. Bane Sullivan[1] **and Alexander A. Kaszynski**[2]

1 Department of Geophysics, Colorado School of Mines, Golden, CO, USA **2** Universal Technology Corporation, Dayton, OH, USA

DOI: 10.21105/joss.01450

Software

- Review
- Repository
- Archive

Submitted: 12 April 2019
Published: 18 May 2019

Summary

There are several options for 3D visualization in Python, a few notable projects include Matplotlib (Hunter, 2007), Mayavi (Ramachandran & Varoquaux, 2011), the yt Project (Turk et al., 2010), and the Visualization Toolkit (VTK) (Schroeder, Lorensen, & Martin, 2006). However, few open-source packages are capable of handling large, spatially-referenced datasets and some are powerful yet have inherently complex application programming interfaces (APIs), which might create a barrier to entry for new users. Notably, VTK is a powerful scientific visualization software library, and with Python bindings, it combines the speed of C++ with the rapid prototyping of Python. Despite this, VTK code programmed in Python using the base VTK Python package is unnecessarily complicated as its API binds existing C++ calls. The PyVista Python package provides a concise, well-documented interface exposing VTK's powerful visualization backend; enabling researchers to rapidly explore large datasets, communicate their spatial findings, and facilitate reproducibility. PyVista further seeks to simplify standard mesh creation and plotting routines without compromising on the speed of the C++ VTK backend.

Plotting VTK datasets using only the VTK Python package or a similar visualization library is often an ambitious programming endeavor. Reading a VTK supported file and plotting it requires a user to write a complicated sequence of routines to render the data object while having to remember which VTK classes to use for file reading and dataset mapping. PyVista includes plotting routines that are intended to be intuitive and highly controllable with syntax and keyword arguments similar to Matplotlib (Hunter, 2007). These plotting routines are defined to make the process of visualizing spatially referenced data straightforward and easily implemented by novice programmers; streamlining the process for creating integrated scenes like that shown in Figure 1.

VTK implements an object-oriented approach to 3D visualization (Schroeder et al., 2006), and PyVista adheres to that underlying structure to provide an API that expands on VTK's data types. These expanded, wrapped types hold methods and attributes for quickly accessing scalar arrays, inspecting properties of the dataset, or using filtering algorithms to transform datasets. PyVista wrapped objects have a suite of common filters ready for immediate use directly on the objects. These filters are commonly used algorithms in the VTK library that have been made more accessible by binding a method to control that algorithm directly onto all PyVista datasets, providing a shared set of functionality. Through the use of these bound filtering methods, powerful VTK algorithms can be leveraged and controlled via keyword arguments designed to be intuitive for novice users.

At its core, PyVista is a pure Python helper module for VTK that interfaces back to VTK data objects through NumPy (Ascher et al., 2001) and direct array access adhering to

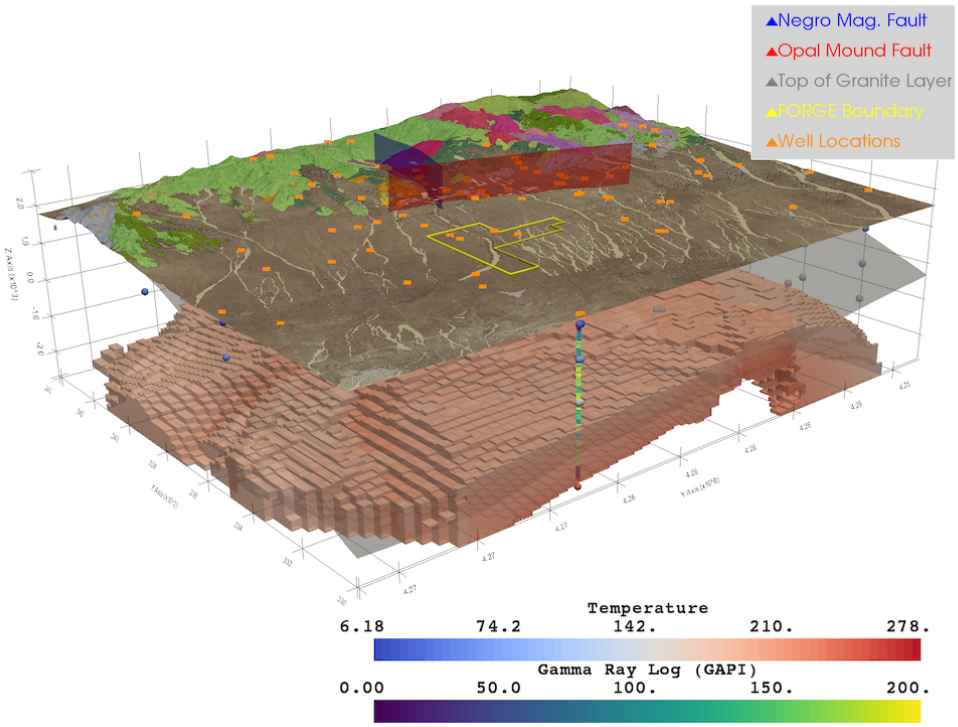

Figure 1: A visually integrated scene of geospatial data (FORGE Geothermal Site). This rendering includes a digital land surface with overlain satellite imagery and geologic map, a subsurface temperature model, scattered points of the sampled temperature values, geophysical well logging data, GIS site boundary, and interpreted faulting surfaces.

VTK's object-oriented approach to 3D visualization (Schroeder et al., 2006). The PyVista Python package provides an accessible and intuitive interface back to the VTK library to facilitate rapid prototyping, analysis, and visual integration of spatially referenced datasets.

Figure 1 demonstrates an integrated scene of geospatial data generated by PyVista; to learn more about how this visualization was created, please visit PVGeo's FORGE project website (forge.pvgeo.org).

Mentions

PyVista is used extensively by the Air Force Research Labs (AFRL) for data visualization and plotting in research articles including figures visualizing 3D tessellated models generated from structured light optical scanner and results from finite element analysis. AFRL publications leveraging PyVista for 3D visualization include: (D. Gillaugh, Kaszynski, Brown, Johnston, & Slater, 2017), (Brown, Beck, Kaszynski, & Clark, 2018a), (Brown, Beck, Kaszynski, & Clark, 2018b), (J. A. Beck, Brown, Kaszynski, & Carper, 2018), (Kaszynski, Beck, & Brown, 2018), (D. L. Gillaugh, Kaszynski, Brown, Johnston, & Slater, 2018), (D. L. Gillaugh, Kaszynski, Brown, Beck, & Slater, 2019)

PVGeo (github.com/OpenGeoVis/PVGeo) is a Python package of VTK-based algorithms to analyze geoscientific data and models. PyVista is used to make the inputs and outputs of PVGeo's algorithms more accessible and to streamline the process of visualizing geoscientific data.

References

Ascher, D., Dubois, P. F., Hinsen, K., Hugunin, J., Oliphant, T., & others. (2001). Numerical python. Citeseer. doi:10.1109/MCSE.2011.37

Beck, J. A., Brown, J. M., Kaszynski, A. A., & Carper, E. B. (2018). Active subspace development of integrally bladed disk dynamic properties due to manufacturing variations. In (p. V07AT32A011). doi:10.1115/GT2018-76800

Brown, J. M., Beck, J., Kaszynski, A., & Clark, J. (2018a). Surrogate modeling of manufacturing variation effects on unsteady interactions in a transonic turbine. *Journal of Engineering for Gas Turbines and Power*, *141*(3), 032506–032506–12. doi:10.1115/1.4041314

Brown, J. M., Beck, J., Kaszynski, A., & Clark, J. (2018b). Surrogate modeling of manufacturing variation effects on unsteady interactions in a transonic turbine. In (p. V07AT32A010). doi:10.1115/GT2018-76609

Gillaugh, D. L., Kaszynski, A. A., Brown, J. M., Beck, J. A., & Slater, J. C. (2019). Mistuning evaluation comparison via as-manufactured models, traveling wave excitation, and compressor rigs. *Journal of Engineering for Gas Turbines and Power*, *141*(6), 061006–061006–13. doi:10.1115/1.4042079

Gillaugh, D. L., Kaszynski, A. A., Brown, J. M., Johnston, D. A., & Slater, J. C. (2018). Accurate strain gauge limits through geometry mistuning modeling. *Journal of Propulsion and Power*, *34*(6), 1401–1408. doi:10.2514/1.B36849

Gillaugh, D., Kaszynski, A., Brown, J. M., Johnston, D. A., & Slater, J. C. (2017). Accurate strain gage limits through geometry mistuning modeling. In AIAA scitech forum. American Institute of Aeronautics; Astronautics. doi:10.2514/6.2017-0865

Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in science & engineering*, *9*(3), 90. doi:10.1109/MCSE.2007.55

Kaszynski, A. A., Beck, J. A., & Brown, J. M. (2018). Automated meshing algorithm for generating as-manufactured finite element models directly from as-measured fan blades and integrally bladed disks. In (p. V07CT35A024). doi:10.1115/GT2018-76375

Ramachandran, P., & Varoquaux, G. (2011). Mayavi: 3D visualization of scientific data. *Computing in Science & Engineering*, *13*(2), 40–51. doi:10.1109/MCSE.2011.35

Schroeder, W. J., Lorensen, B., & Martin, K. (2006). *The visualization toolkit (4th ed.)*. Kitware.

Turk, M. J., Smith, B. D., Oishi, J. S., Skory, S., Skillman, S. W., Abel, T., & Norman, M. L. (2010). Yt: A multi-code analysis toolkit for astrophysical simulation data. *The Astrophysical Journal Supplement Series*, *192*(1), 9. doi:10.1088/0067-0049/192/1/9