prysm: A Python optics module

Brandon Dube[1]

1 Retro Refractions, LLC

Summary

`prysm` is an open-source library for physical and first-order modeling of optical systems and analysis of related data.

The library supports the modeling, evaluation, and visualization of an optical system at any level, in the pupil plane, image plane, or k-space (MTF). This can also be done at the optical, or opto-electronic system level by including detector elements. Optical propagations are handled using the paraxial approximation of the Fresnel Transform via an FFT implementation.

`prysm` also features an `io` submodule for loading data from commercial and open source instrumentation and software into simple python structures (dicts, etc) or static methods on its classes for loading directly into `prysm`'s object system. Notably, this support includes the most popular interferometers and MTF benches in the commercial marketplace. Combined, these capabilities serve as the backbone of user programs supporting imaging system analysis by performing tasks such as image simulation, wavefront sensing, or robust analysis of metrology data utilizing cutting-edge methods from the literature.

The library is available for Linux, MacOS, and Windows and only carries core dependencies on numpy (T. E. Oliphant, 2006) and scipy (Jones, Oliphant, Peterson, & others, 2001). It will utilize a wide array of optional dependencies for some functionality. For performance, prysm can leverage numba (Lam, Pitrou, & Seibert, 2015) for acceleration of calculations on CPUs or cupy (Nishino & Loomis, 2017) on GPUs. Plotting is implemented using matplotlib (Hunter, 2007), images are read and written using imageio (Klein et al., 2019) and deconvolved with point spread functions using scikit-image (Van der Walt et al., 2014). Some tabular formats for MTF data require pandas (McKinney, 2010). Reading of Zygo's datx format for interferometric data requires h5py (Collette, 2013).

Acknowledgements

References

Collette, A. (2013). *Python and hdf5.* O'Reilly.

Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing In Science & Engineering*, *9*(3), 90–95. doi:10.1109/MCSE.2007.55

Jones, E., Oliphant, T., Peterson, P., & others. (2001). SciPy: Open source scientific tools for Python. Retrieved from http://www.scipy.org/

Klein, A., Silvester, S., Tanbakuchi, A., Müller, P., Nunez-Iglesias, J., Harfouche, M., McCormick, M., et al. (2019, February). Imageio/imageio: V2.5.0. doi:10.5281/zenodo.2558175

Lam, S. K., Pitrou, A., & Seibert, S. (2015). Numba: A llvm-based python jit compiler. In *Proceedings of the second workshop on the llvm compiler infrastructure in hpc* (p. 7). ACM.

McKinney, W. (2010). Data structures for statistical computing in python. In S. van der Walt & J. Millman (Eds.), *Proceedings of the 9th python in science conference* (pp. 51–56).

Nishino, R. O. Y. U. D., & Loomis, S. H. C. (2017). CuPy: A numpy-compatible library for nvidia gpu calculations. In *31st confernce on neural information processing systems*.

Oliphant, T. E. (2006). *A guide to numpy* (Vol. 1).

Van der Walt, S., Schönberger, J. L., Nunez-Iglesias, J., Boulogne, F., Warner, J. D., Yager, N., Gouillart, E., et al. (2014). Scikit-image: Image processing in python. *PeerJ*, *2*, e453.