The Journal of Open Source Software

Scientific Computational Imaging Code (SCICO)

Thilo Balke [1,2], **Fernando Davis** [1,3], **Cristina Garcia-Cardona** [1], **Soumendu Majee** [4], **Michael McCann** [1], **Luke Pfister** [1], **and Brendt Wohlberg** [1]

1 Los Alamos National Laboratory **2** Purdue University **3** University of Puerto Rico-Mayaguez **4** Samsung Research America

DOI: 10.21105/joss.04722

Software

- Review
- Repository
- Archive

Editor: Daniel S. Katz

Reviewers:

- @vitorsr
- @DanNixon
- @lucaferranti

Submitted: 19 August 2022
Published: 28 October 2022

Summary

Scientific Computational Imaging Code (SCICO) is a Python package for solving the inverse problems that arise in scientific imaging applications. Its primary focus is providing methods for solving ill-posed inverse problems by using an appropriate prior model of the reconstruction space. SCICO includes a growing suite of operators, cost functionals, regularizers, and optimization routines that may be combined to solve a wide range of problems, and is designed so that it is easy to add new building blocks. SCICO is built on top of JAX rather than NumPy, enabling GPU/TPU acceleration, just-in-time compilation, and automatic gradient functionality, which is used to automatically compute the adjoints of linear operators. An example of how to solve a multi-channel tomography problem with SCICO is shown in Figure 1. The SCICO source code is available from GitHub, and pre-built packages are available from PyPI. It has extensive online documentation, including API documentation and usage examples, which can be run online at Google Colab and binder.

Community contributions, including bug reports, feature requests, and code contributions, are welcomed at https://github.com/lanl/scico.

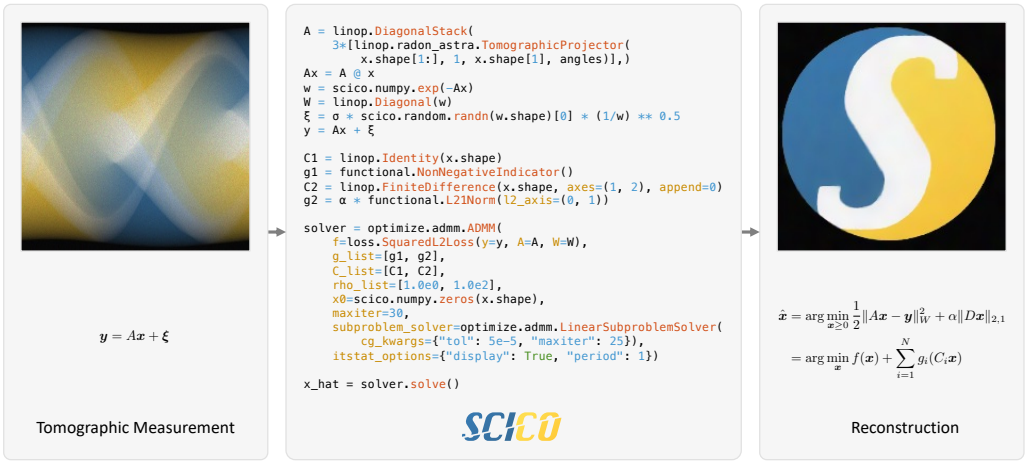

Figure 1: Solving a multi-channel tomography problem with SCICO.

Statement of Need

In traditional imaging, the burden of image formation is placed on physical components, such as a lens, with the resulting image being taken from the sensor with minimal processing. In computational imaging, in contrast, the burden of image formation is shared with or shifted

to computation, with the resulting image typically being very different from the measured data. Common examples of computational imaging include demosaicing in consumer cameras, computed tomography and magnetic resonance imaging in medicine, and synthetic aperture radar in remote sensing. This is an active and growing area of research, and many of these problems have common properties that could be supported by shared implementations of solution components.

The goal of SCICO is to provide a general research tool for computational imaging, with a particular focus on scientific imaging applications, which are particularly underrepresented in the existing range of open-source packages in this area. While a number of other packages overlap somewhat in functionality with SCICO, only a few support execution of the same code on both CPU and GPU devices, and we are not aware of any that support just-in-time compilation and automatic gradient computation, which is invaluable in computational imaging. SCICO provides all three of these valuable features by being built on top of JAX rather than NumPy.

Solving Imaging Inverse Problems in SCICO

SCICO provides a set of building blocks that can be used to express a wide variety of problems and their corresponding solutions. These building blocks include operators for representing the *forward model* of an imaging problem, functionals for representing *data fidelity* and *regularization* terms, and optimization algorithms for minimizing these functionals. The online documentation includes a guide to the use of these components as well as numerous example scripts demonstrating their use in practice.

Machine Learning

SCICO includes an implementation of the DnCNN denoiser (Zhang et al., 2017), which can be applied to other inverse problems via the *plug-and-play priors* (PPP) (Kamilov et al., 2022; Sreehari et al., 2016; Venkatakrishnan et al., 2013) framework (see Figure 2). A number of other leading machine learning methods have been implemented, and are expected to be merged into the main SCICO github branch in the near future.

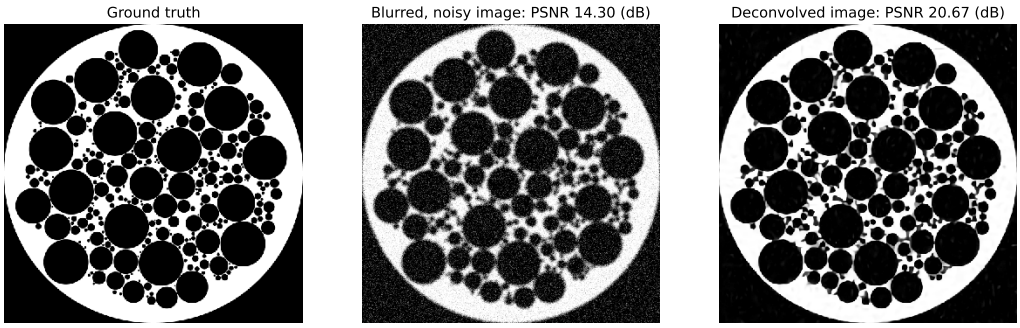

Figure 2: Image deconvolution via PPP with DnCNN denoiser.

Advantages of JAX-based Design

The vast majority of scientific computing packages in Python are based on NumPy and SciPy. SCICO, in contrast, is based on JAX, which provides most of the same features, but with the addition of automatic differentiation, GPU support, and just-in-time (JIT) compilation. In addition to its obvious application in gradient-based minimization methods, automatic differentiation allows automatic computation of the adjoint operator of a linear operator, the manual derivation of which is often time-consuming.

Acknowledgments

This work was supported by the Laboratory Directed Research and Development program of Los Alamos National Laboratory under project number 20200061DR.

References

Kamilov, U. S., Bouman, C. A., Buzzard, G. T., & Wohlberg, B. (2022). Plug-and-play methods for integrating physical and learned models in computational imaging. To appear in *IEEE Signal Processing Magazine*. https://doi.org/10.48550/arXiv.2203.17061

Sreehari, S., Venkatakrishnan, S. V., Wohlberg, B., Buzzard, G. T., Drummy, L. F., Simmons, J. P., & Bouman, C. A. (2016). Plug-and-play priors for bright field electron tomography and sparse interpolation. *IEEE Transactions on Computational Imaging*, *2*(4), 408–423. https://doi.org/10.1109/TCI.2016.2599778

Venkatakrishnan, S. V., Bouman, C. A., & Wohlberg, B. (2013). Plug-and-play priors for model based reconstruction. *Proceedings of IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, 945–948. https://doi.org/10.1109/GlobalSIP.2013.6737048

Zhang, K., Zuo, W., Chen, Y., Meng, D., & Zhang, L. (2017). Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising. *IEEE Transactions on Image Processing*, *26*(7), 3142–3155. https://doi.org/10.1109/TIP.2017.2662206