

Derotation: a Python package for correcting rotation-induced distortions in line-scanning microscopy

Laura Porta¹, Simon Weiler¹, Adam L. Tyson², and Troy W. Margrie¹

¹ The Sainsbury Wellcome Centre for Neural Circuits and Behaviour, University College London, 25 Howland Street, London W1T 4JG, UK ² Neuroinformatics Unit, Sainsbury Wellcome Centre & Gatsby Computational Neuroscience Unit, University College London, London W1T 4JG, UK

DOI: [DOIunavailable](#)

Software

- [Review](#)
- [Repository](#)
- [Archive](#)

Editor: [Pending Editor](#)

Reviewers:

- [@Pending Reviewers](#)

Submitted: N/A

Published: N/A

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](#)).

Summary

In line-scanning microscopy, sample rotation during acquisition introduces geometric distortions that compromise downstream analyses such as cell detection and signal extraction. While several groups have developed custom solutions (Hennestad et al., 2021; Sit & Goard, 2023; Vélez-Fort et al., 2018; Voigts & Harnett, 2020), a general-purpose tool has been lacking. *derotation* is an open-source Python package that corrects these artifacts by applying a line-by-line inverse rotation using recorded angles and the microscope's line acquisition clock, restoring the expected geometry and enabling reliable cell segmentation during rapid rotational movements (Figure 1).

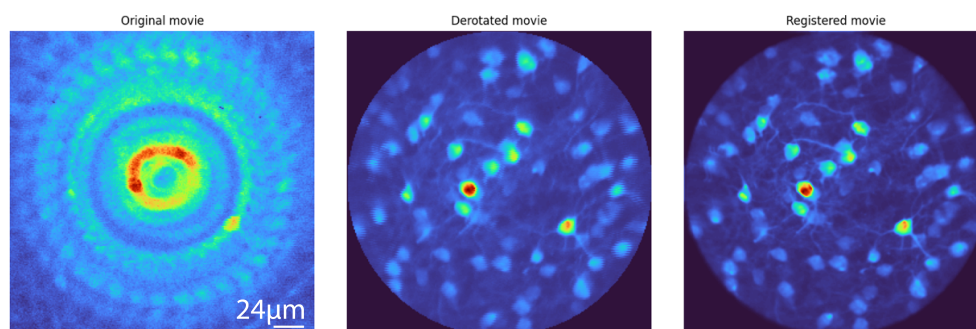


Figure 1: Example of derotation correction. On the left, the average of a series of images acquired using 3-photon microscopy of layer 6 mouse cortical neurons labeled with GCaMP7f during passive rotation. Center, the mean image after derotation, and on the right the mean image of the derotated movie after suite2p registration (Pachitariu et al., 2016). As you can see, following derotation the cells are visible and have well defined shapes.

Statement of Need

When sample motion during acquisition is rotational, it produces characteristic “fan-like” distortions that corrupt the morphological features of the imaged structures (Figure 2), significantly complicating or preventing cell segmentation and automated region-of-interest tracking.

This problem is particularly acute in systems neuroscience, where researchers increasingly combine multiphoton calcium imaging with behavioral paradigms involving head rotation (Hennestad et al., 2021; Sit & Goard, 2023; Vélez-Fort et al., 2018; Voigts & Harnett, 2020). High-speed angular motion can render imaging data unusable, especially for modalities with lower frame rates such as three-photon imaging. Despite multiple lab-specific solutions, no validated, open-source, and easy-to-use Python tool has been available to the broader community.

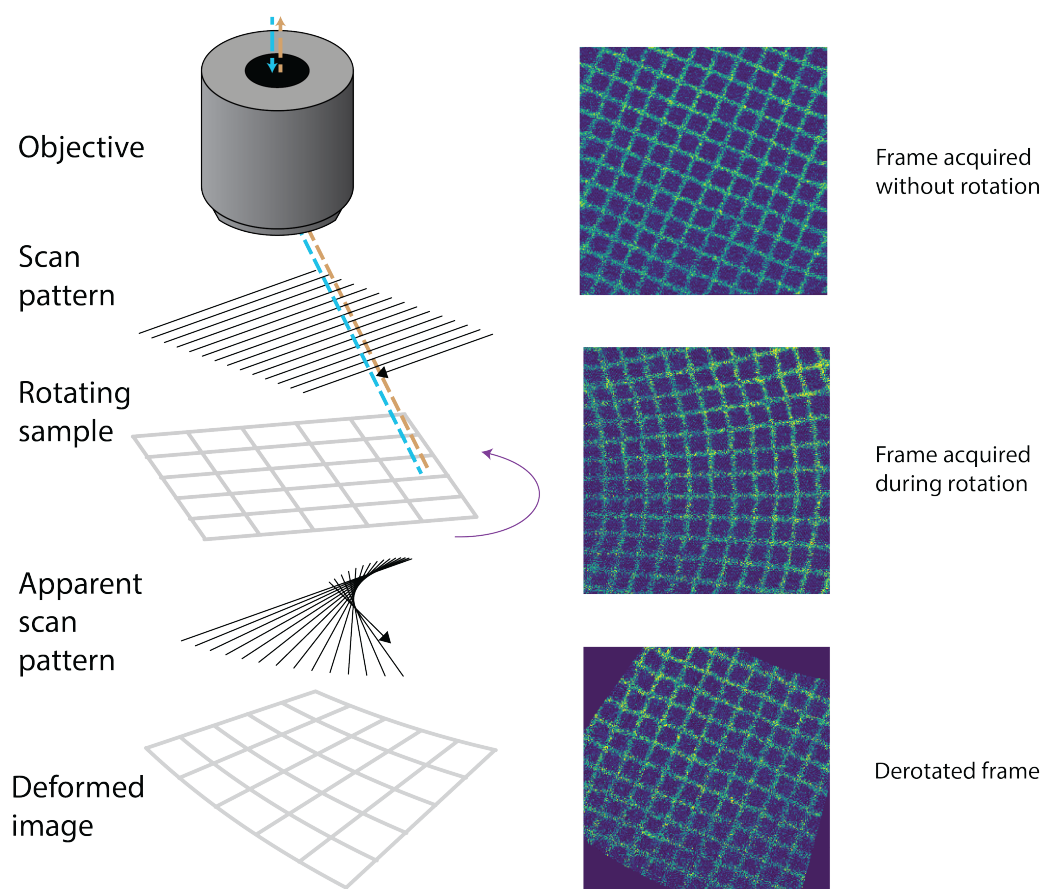


Figure 2: Schematic of line-scanning microscope distortion. Left: line scanning pattern plus sample rotation lead to fan-like artifacts when imaging a grid. Right: grid imaged while still (top), while rotating at $200^\circ/\text{s}$ with 7Hz frame rate (middle), and after derotation (bottom), showing alignment restoration.

derotation meets this need by providing a documented, tested, and modular solution for post hoc correction of imaging data acquired during rotation. The package has been developed openly on GitHub since June 2023, with contributions from five developers. It is distributed on PyPI under a BSD-3-Clause license, with comprehensive documentation, tutorials, and runnable examples via Binder at <https://derotation.neuroinformatics.dev>. By providing a robust and accessible tool, derotation lowers the barrier for complex behavioral experiments and improves the reproducibility of a key analysis step in a growing field of research.

The package has been validated on three-photon recordings of deep cortical neurons expressing the calcium indicator GCaMP7f in head-fixed mice (Figure 3). The corrected images showed restored cellular morphology and were successfully processed by Suite2p (Pachitariu et al., 2016). Compared with frame-by-frame affine correction, line-by-line derotation preserves ROI fluorescence signals during rotation periods, eliminating the artificial dips visible in Figure 3.

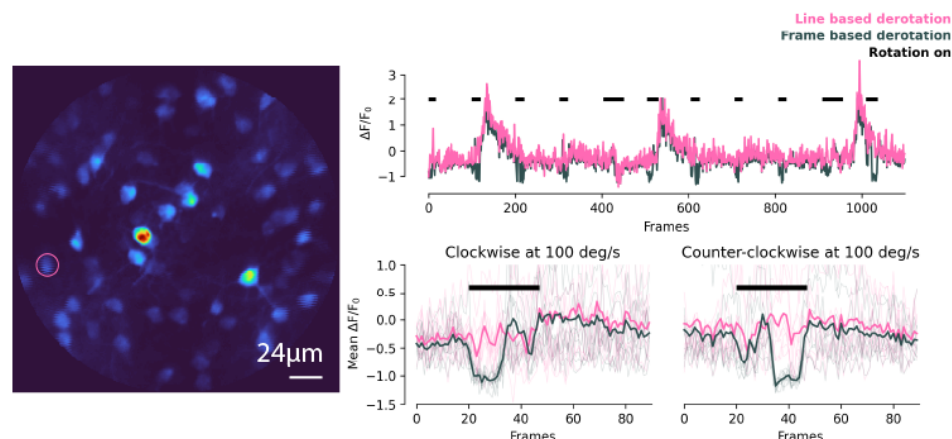


Figure 3: Figure 3. Validation on 3-photon data. Left: mean image after line-by-line derotation. Red circle marks the ROI used for the plots on the right. Top right: sample $\Delta F/F_0$ timecourse for the selected ROI (pink = line-by-line derotation; gray = frame-by-frame affine correction; shaded vertical bars = rotation intervals). Bottom right: mean $\Delta F/F_0$ aligned to rotation periods for clockwise and counterclockwise rotations. Line-by-line derotation preserves the ROI signal during rotations and removes the artificial dips introduced by frame-by-frame correction. Clockwise and counterclockwise traces show a roughly mirror-symmetric, angle-dependent modulation of measured fluorescence with the frame-by-frame correction.

State of the Field

Several groups have developed rotation correction procedures as part of their experimental pipelines (Hennestad et al., 2021; Sit & Goard, 2023; Vélez-Fort et al., 2018), but each solution remains embedded in a lab-specific workflow, typically applying frame-level corrections in MATLAB. The closest prior work is Voigts & Harnett (2020), who described a line-by-line derotation approach for two-photon imaging during free locomotion, but their code was not designed to generalize across setups with lower frame rates. General-purpose image registration tools such as Suite2p (Pachitariu et al., 2016) correct for translational motion but cannot handle within-frame rotational distortions. `derotation` fills this gap by integrating with the scientific Python ecosystem and producing output that can be directly fed into standard registration pipelines.

Software Design

`derotation` separates the core algorithm from experiment-specific logic through a layered, object-oriented design. The core function (`derotate_an_image_array_line_by_line`) takes a movie and a per-line angle array and returns the corrected stack, with no dependencies on configuration or I/O, so it can be used directly in any custom workflow.

On top of this core, two pipeline classes (`FullPipeline` and `IncrementalPipeline`) orchestrate the end-to-end processing: data loading, angle interpolation, optional center-of-rotation estimation, derotation, and output saving. Each processing step is implemented as an overridable method, so users can subclass either pipeline to adapt to new experimental setups or data formats. A simulation module generates synthetic test data for systematic regression testing. Full API documentation, configuration guides, and runnable examples are available at <https://derotation.neuroinformatics.dev>.

Research Impact Statement

`derotation` is in use at the Sainsbury Wellcome Centre to process three-photon calcium imaging data acquired during passive head rotation experiments, with its output fed into Suite2p (Pachitariu et al., 2016) for cell detection and signal extraction.

AI Usage Disclosure

GitHub Copilot was used for code autocompletion during development and to assist with drafting portions of this manuscript and the documentation. All AI-generated content was reviewed, tested, and validated by the authors, who carried out the algorithmic design, architectural decisions, and scientific validation.

Acknowledgements

We thank Eivind Hennestad for initial project discussion. We thank Mateo Vélez-Fort and Chryssanthi Tsitoura for their assistance in building and testing the the three-photon imaging and rotation setup as well as for feedback on the package. We also thank Igor Tatarnikov for contributing to the development of the package and the whole Neuroinformatics Unit. This package was inspired by previous work on `derotation` as described in (Voigts & Harnett, 2020). The authors are grateful to the support staff of the Neurobiological Research Facility at the Sainsbury Wellcome Centre (SWC). This research was funded by the Sainsbury Wellcome Centre core grant from the Gatsby Charitable Foundation (GAT3361) and Wellcome Trust (219627/Z/19/Z), a Wellcome Trust Investigator Award (214333/Z/18/Z) and Discovery Award (306384/Z/23/Z) to T.W.M. and by SWC core funding to the Neurobiological Research Facility. S.W. was funded by a Feodor-Lynen fellowship from the Alexander von Humboldt Foundation.

References

- Hennestad, E., Witoelar, A., Chambers, A. R., & Vervaeke, K. (2021). Mapping vestibular and visual contributions to angular head velocity tuning in the cortex. *Cell Reports*, 37(12), 110134. <https://doi.org/10.1016/j.celrep.2021.110134>
- Pachitariu, M., Stringer, C., Dipoppa, M., Schröder, S., Rossi, L. F., Dalgleish, H., Carandini, M., & Harris, K. D. (2016). *Suite2p: Beyond 10,000 neurons with standard two-photon microscopy*. Neuroscience. <https://doi.org/10.1101/061507>
- Sit, K. K., & Goard, M. J. (2023). Coregistration of heading to visual cues in retrosplenial cortex. *Nature Communications*, 14(1), 1992. <https://doi.org/10.1038/s41467-023-37704-5>
- Vélez-Fort, M., Bracey, E. F., Keshavarzi, S., Rousseau, C. V., Cossell, L., Lenzi, S. C., Strom, M., & Margrie, T. W. (2018). A Circuit for Integration of Head- and Visual-Motion Signals in Layer 6 of Mouse Primary Visual Cortex. *Neuron*, 98(1), 179–191.e6. <https://doi.org/10.1016/j.neuron.2018.02.023>
- Voigts, J., & Harnett, M. T. (2020). Somatic and Dendritic Encoding of Spatial Variables in Retrosplenial Cortex Differs during 2D Navigation. *Neuron*, 105(2), 237–245.e4. <https://doi.org/10.1016/j.neuron.2019.10.016>