

dfreproject: A Python package for astronomical reprojection

Carter Lee Rhea^{1,2*}, Pieter Van Dokkum^{1,3*}, Steven R. Janssens¹, Imad Pasha^{1,3}, Roberto Abraham^{1,4,5}, William P Bowman^{1,3}, Deborah Lokhorst^{1,6}, and Seery Chen^{1,4,5}

¹ Dragonfly Focused Research Organization, 150 Washington Avenue, Santa Fe, 87501, NM, USA ² Centre de recherche en astrophysique du Québec (CRAQ) ³ Astronomy Department, Yale University, 219 Prospect St, New Haven, CT 06511, USA ⁴ David A. Dunlap Department of Astronomy & Astrophysics, University of Toronto, 50 St. George Street, Toronto, ON M5S3H4, Canada ⁵ Dunlap Institute for Astronomy & Astrophysics, University of Toronto, 50 St. George Street, Toronto, ON M5S3H4, Canada ⁶ NRC Herzberg Astronomy & Astrophysics Research Centre, 5071 West Saanich Road, Victoria, BC V9E 2E7, Canada * These authors contributed equally.

DOI: [10.xxxxxx/draft](https://doi.org/10.xxxxxx/draft)

Software

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

Editor: Axel Donath

Reviewers:

- [@svank](#)
- [@keflavich](#)

Submitted: 07 April 2025

Published: unpublished

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](https://creativecommons.org/licenses/by/4.0/))

Summary

Deep astronomical images are often constructed by digitally stacking many individual sub-exposures. Each sub-exposure is expected to show small differences in the positions of stars and other objects in the field, due to the movement of the celestial bodies, changes/imperfections in the opto-mechanical imaging train, and other factors. To maximize image quality, one must ensure that each sub-exposure is aligned to a common frame of reference prior to stacking. This is done by reprojecting each exposure onto a common target grid defined using a World Coordinate System (WCS) that is defined by mapping the known angular positions of reference objects to their observed spatial positions on each image. The transformations needed to reproject images involve complicated trigonometric expressions which can be slow to compute, so reprojection can be a major bottleneck in image processing pipelines.

To make astronomical reprojections faster to implement in pipelines, we have written dfreproject, a Python package of GPU-optimized functions for this purpose. The package's functions break down coordinate transformations using Gnomonic projections to define pixel-by-pixel shifts from the source to the target plane. The package also provides tools for interpolating a source image onto a target plane with a single function call. This module follows the FITS and SIP formats laid out in the following papers: Greisen & Calabretta (2002), Calabretta & Greisen (2002), and Shupe et al. (2005). Compared to common alternatives, dfreproject's routines result in speedups of up to 20X when run on a GPU and 10X when run on a CPU.

Statement of need

Several packages already exist for calculating and applying the reprojection of a source image onto a target plane, such as reproject (Robitaille et al., 2020) or astroalign (Beroiz et al., 2020). These packages excel at easy-to-use, general-purpose astronomical image reprojection, but they function solely on CPUs and can be computational bottlenecks in some data reduction pipelines. The dfreproject package harnesses GPUs (using PyTorch (Paszke et al., 2019) as its computational backbone) to improve computational efficiency. The package has minimal reliance on pre-existing WCS packages such as those found in astropy or WCSLIB (Astropy Collaboration et al., 2013, 2018, 2022), with such packages being used only for convenience in

non-computationally expensive steps (such as using `astropy.wcs` to read the header information from the input fits files).

`dfreproject`'s primary purpose is to reproject observations taken by a new version of the Dragonfly Telephoto Array that is currently under construction in Chile. The volume of data obtained with this telescope will be large, with >1000 exposures obtained simultaneously, and it is paramount that the processing pipeline incorporates fast and accurate reprojection methods. As shown below, by leveraging PyTorch for vectorization and parallelization via the GPU, `dfreproject` achieves a considerable speedup (up to nearly 20X) over standard methods.

`dfreproject` can be used as a direct replacement for `reproject.reproject_interp` by simply importing `dfreproject` instead of `reproject` such as:

```
from dfreproject import calculate_reprojection
reprojected = calculate_reprojection(
    source_hdus=source_hdu,
    target_wcs=target_wcs,
    shape_out=target_hdu.data.shape,
    order='bilinear',
    device='cpu',
    n_threads=2
)
```

The `target_wcs` argument can be passed as a header similar to `reproject`. Additionally, if `shape_out` is not provided, the shape will be the same as the input.

Methods

We must perform three intermediate calculations to reproject an image onto a new coordinate plane. To do this, we use the target and source WCS.

Before defining the steps, there are a few terms to define:

- **SIP:** Standard Imaging Polynomial. This convention allows us to represent non-linear geometric distortions as a simple polynomial. The order and coefficients of this polynomial are stored in the header. The SIP is broken down into four individual polynomials, `SIP_A`, `SIP_B`, `SIP_INV_A`, and `SIP_INV_B` where `SIP_A` defines the polynomial applied to the x-coordinates, `SIP_B` defines the polynomial applied to the y-coordinates, and `SIP_INV_A` and `SIP_INV_B` define the inverse operations. For an in-depth discussion on SIP, please see Shupe et al. (2005).

CD Matrix: Coordinate Description Matrix. This is a 2X2 matrix that encodes the image's rotation, skew, and scaling. The values are conveniently stored in the header. The CD matrix may also be constructed from the PC Projection Coordinate matrix multiplied by the CDELTA values.

The steps are as follows:

1. For each pixel, calculate the corresponding celestial coordinate using the target WCS
 1. Apply shift
 2. Apply SIP distortion
 3. Apply CD matrix
 4. Apply transformation to celestial coordinates using the Gnomonic projection
2. Calculate the position in the source grid for each celestial coordinate. This provides the offset for the next step.
 1. Apply the inverse transformation using the Gnomonic projection

78 2. Apply inverse CD matrix
79 3. Apply inverse SIP distortion
80 4. Apply shift
81 3. Interpolate the source image onto the newly calculated grid
82 In the final interpolation step, we include local flux conservation by simultaneously projecting an
83 identity tensor called the footprint. The final reprojected frame is normalized by this footprint.

84 Results

85 Demo

86 We created two small (50x50) FITS files for this demonstration with a pixel offset of 0.5 pixels
87 and a 0.005 degree offset in the header. In Figure 1 from left to right, we show the original
88 image, the dfreproject solution, the reproject solution, and the relative error between the
89 two. We define the relative error as $100 * \frac{\text{dfreproject_solution} - \text{reproject_solution}}{\text{reproject_solution}}$. For both solutions,
90 we use a bilinear interpolation scheme. In the noisy regions of the image, the differences in
91 the reprojections is pure noise. There are slight differences in the solutions at the locations of
92 the Gaussians which is attributable to small differences in the normalization. In this demo, we
93 used the gpu by setting device='gpu' in the argument of calculate_reprojection().

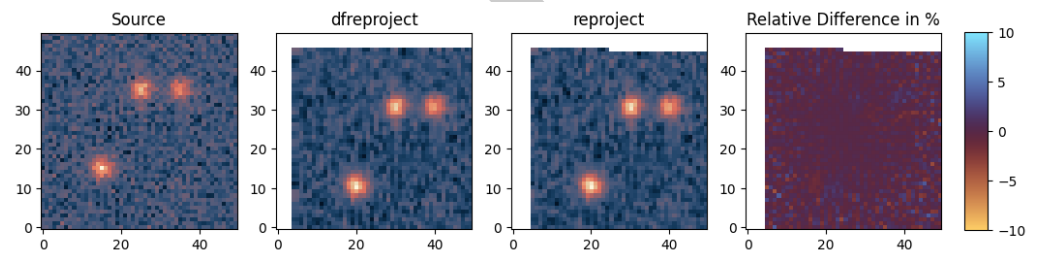


Figure 1

94 Speed Comparison

95 To compare the execution times, we created a benchmarking script (which can be found in
96 the demos/benchmarking directory under benchmark-script.py; the figures are constructed
97 with benchmark-plotter.py). This test is run between dfreproject and reproject. We
98 benchmark the three interpolation schemes with and without SIP distortion for images sized
99 256x256 to 4000x6000 (this approximately matches the size of Dragonfly images). Figure 2
100 shows the results of this benchmarking when dfreproject is run using a GPU (NVIDIA
101 GeForce RTX 4060).

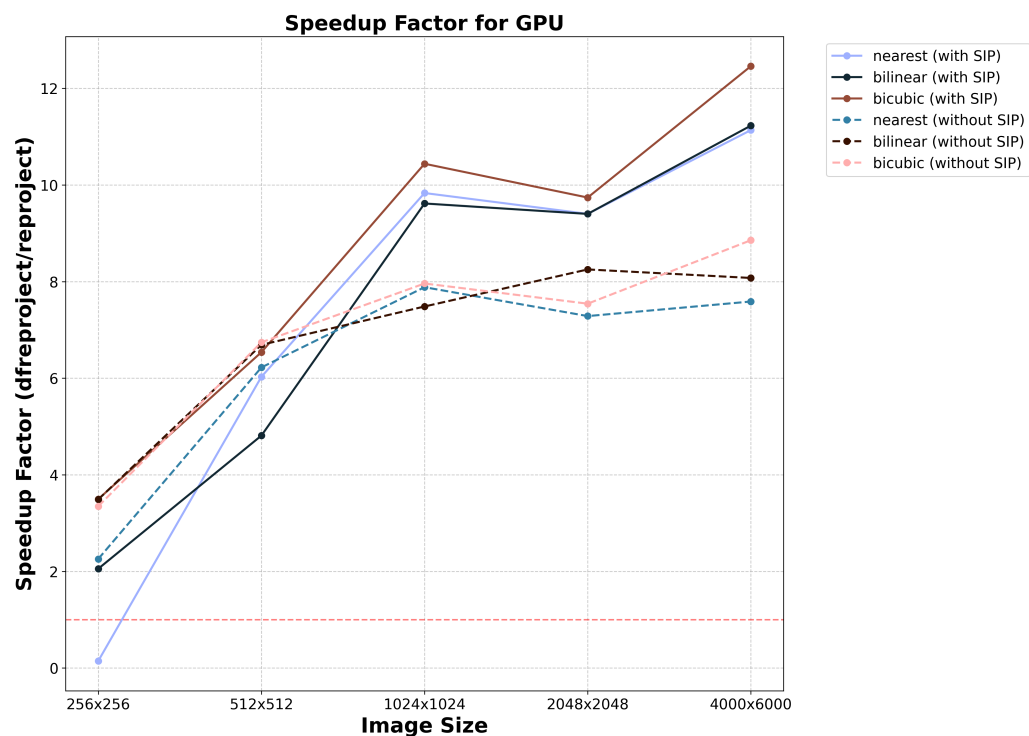


Figure 2

As evidenced by this figure, dfreproject has a significant speed advantage over reproject for larger images regardless of the type of interpolation scheme. The speedup is most pronounced in the case where SIP distortions are included.

In Figure 3, we display the same results except we used a CPU (Intel® Core™ i9-14900HX). For this experiment, we used a single thread.

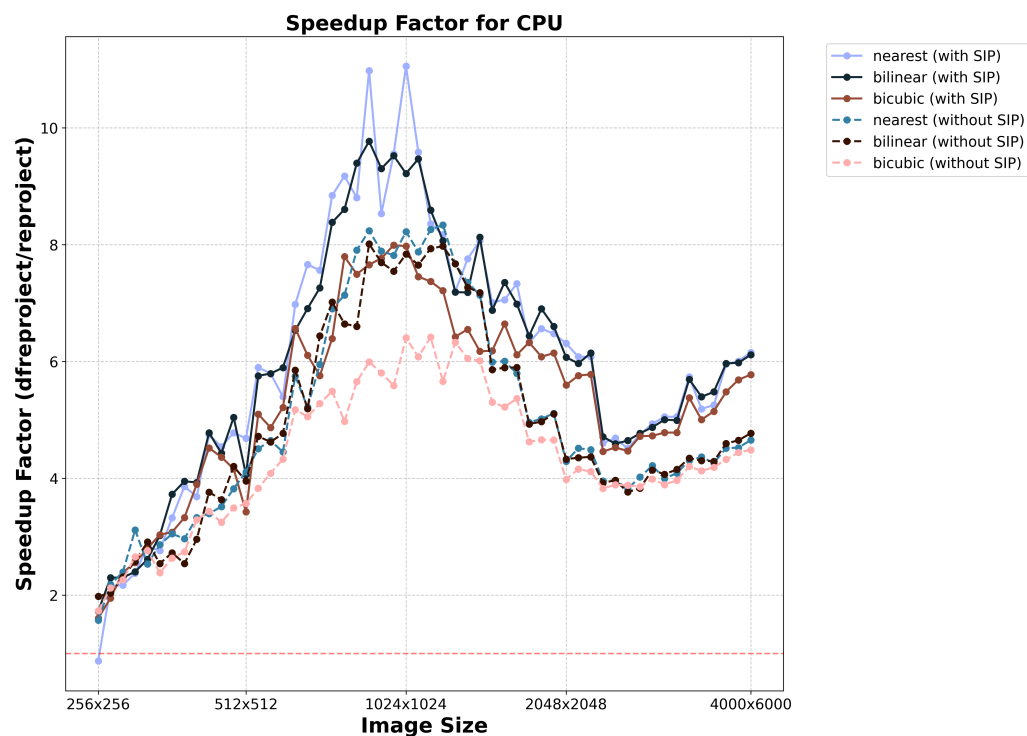


Figure 3

107 Although the speedup on the CPU is not as impressive as on the GPU, it is still considerable.

108 All code can be found in the demo directory.

109 Acknowledgements

110 We acknowledge the Dragonfly FRO and particularly thank Lisa Sloan for her project manage-
111 ment skills.

112 We use the cmcrameri scientific color maps in our demos (Crameri, 2023).

113 References

- 114 Astropy Collaboration, Price-Whelan, A. M., Lim, P. L., Earl, N., Starkman, N., Bradley, L.,
115 Shupe, D. L., Patil, A. A., Corrales, L., Brasseur, C. E., Nöthe, M., Donath, A., Tollerud,
116 E., Morris, B. M., Ginsburg, A., Vaher, E., Weaver, B. A., Tocknell, J., Jamieson, W., ...
117 Astropy Project Contributors. (2022). The Astropy Project: Sustaining and Growing a
118 Community-oriented Open-source Project and the Latest Major Release (v5.0) of the Core
119 Package. *The Astrophysical Journal*, 935, 167. <https://doi.org/10.3847/1538-4357/ac7c74>
- 120 Astropy Collaboration, Price-Whelan, A. M., Sipőcz, B. M., Günther, H. M., Lim, P. L.,
121 Crawford, S. M., Conseil, S., Shupe, D. L., Craig, M. W., Dencheva, N., Ginsburg, A.,
122 VanderPlas, J. T., Bradley, L. D., Pérez-Suárez, D., Val-Borro, M. de, Aldcroft, T. L.,
123 Cruz, K. L., Robitaille, T. P., Tollerud, E. J., ... Astropy Contributors. (2018). The Astropy
124 Project: Building an Open-science Project and Status of the v2.0 Core Package. *The*
125 *Astronomical Journal*, 156, 123. <https://doi.org/10.3847/1538-3881/aabc4f>
- 126 Astropy Collaboration, Robitaille, T. P., Tollerud, E. J., Greenfield, P., Droettboom, M., Bray,
127 E., Aldcroft, T., Davis, M., Ginsburg, A., Price-Whelan, A. M., Kerzendorf, W. E., Conley,

- 128 A., Crighton, N., Barbary, K., Muna, D., Ferguson, H., Grollier, F., Parikh, M. M., Nair,
129 P. H., ... Streicher, O. (2013). Astropy: A community Python package for astronomy.
130 *Astronomy and Astrophysics*, 558, A33. <https://doi.org/10.1051/0004-6361/201322068>
- 131 Beroiz, M., Cabral, J. B., & Sanchez, B. (2020). Astroalign: A Python module for astronomical
132 image registration. *Astronomy and Computing*, 32, 100384. [https://doi.org/10.1016/j.
133 ascom.2020.100384](https://doi.org/10.1016/j.ascom.2020.100384)
- 134 Calabretta, M. R., & Greisen, E. W. (2002). Representations of celestial coordinates in FITS.
135 *A&A*, 395(3), 1077–1122. <https://doi.org/10.1051/0004-6361:20021327>
- 136 Crameri, F. (2023). *Scientific colour maps*. Zenodo. <https://doi.org/10.5281/zenodo.8409685>
- 137 Greisen, E. W., & Calabretta, M. R. (2002). Representations of world coordinates in FITS. *As-*
138 *tronomy and Astrophysics*, 395, 1061–1075. <https://doi.org/10.1051/0004-6361:20021326>
- 139 Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z.,
140 Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M.,
141 Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., ... Chintala, S. (2019). *PyTorch: An*
142 *Imperative Style, High-Performance Deep Learning Library*. [https://doi.org/10.48550/
143 arXiv.1912.01703](https://doi.org/10.48550/arXiv.1912.01703)
- 144 Robitaille, T., Deil, C., & Ginsburg, A. (2020). Reproject: Python-based astronomical image
145 reprojection. *Astrophysics Source Code Library*, ascl:2011.023. [https://ui.adsabs.harvard.
146 edu/abs/2020ascl.soft11023R](https://ui.adsabs.harvard.edu/abs/2020ascl.soft11023R)
- 147 Shupe, D. L., Moshir, M., Li, J., Makovoz, D., Narron, R., & Hook, R. N. (2005). *The*
148 *SIP Convention for Representing Distortion in FITS Image Headers*. 347, 491. [https:
149 //ui.adsabs.harvard.edu/abs/2005ASPC..347..491S](https://ui.adsabs.harvard.edu/abs/2005ASPC..347..491S)