# Characterizing Solar Wind Dynamics in Coronagraph Data

Jake R. Wilson (University of Maryland)
Mentors: Craig DeForest (SwRI), Barbara Thompson(NASA)
Jakewilson0314@gmail.com

## ABSTRACT

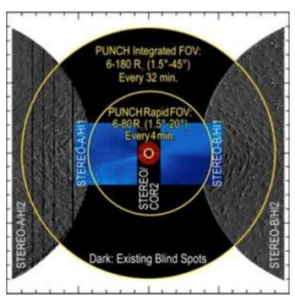
The solar corona is the source of the solar wind that fills our solar system. The flow of the solar wind is far from steady, and noise-reduced coronagraph images from NASA's STEREO mission show that the outer corona is a riotous torrent of visible features rather than a smooth flow. NASA's upcoming mission PUNCH (Polarimeter to UNify the Corona and Heliosphere) will examine the young solar wind, which is the region from the middle of the solar corona to 1AU. PUNCH will be able to study how the solar wind evolves, allowing an unification of the Corona and Heliosphere.

This project focused on creating a Python data pipeline that will process the data coming out of PUNCH. The prototype data pipeline already exists in Perl Data Language (PDL). However, PDL lacks support from the broader data science and heliophysics fields, leading it to be ported into the more mainstream language of Python. In order to port the data pipeline into Python, PDL Transform also had to be ported into Python 3. PDL Transform is a library in PDL that takes in images or data and applies linear and non-linear transforms to them. Unlike methods that already exist in Python, this allows for more freedom in what kind of transform you want to apply to your data.

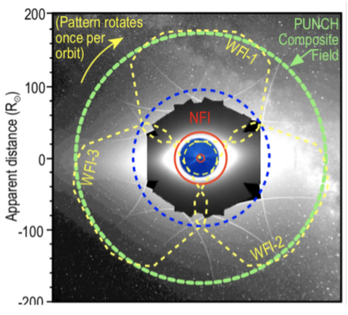
In the end, this project successfully ported over PDL Transform and laid the groundwork for finishing the Python data pipeline.

## 1) PUNCH (Polarimeter to UNify the Corona and Heliosphere) and STEREO COR2 Deep Exposure Campaign

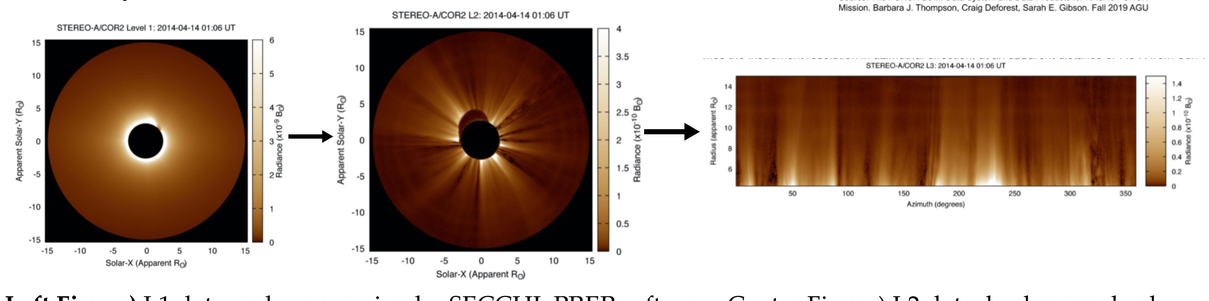
**Top Right Figure)** Consists of one Near Field Imager (NFI), and three Wide Field Imagers(WFI). The WFIs rotate once every orbit, allowing them to create a composite field by creating a mosiac from the images.
**Top Left Figure)** Showing the two FOVs of the NFI and WFIs, and comparing them what we currently have.




Source: Gibson-PUNCH-AsiaPacific Presentation




Source: The PUNCH Bowl: Data System and Data Products for NASA's PUNCH Mission. Barbara J. Thompson, Craig Deforest, Sarah E. Gibson. Fall 2019 AGU




**Left Figure)** L1 data, only processing by SECCHI_PREP software. Center Figure) L2 data, background subtraction, and derotation. Right Figure) L3 data, transformed from 2048x2048 pixels resampled into 3600x800 pixels in polar coordinates. All of this code written in PDL, and needs to be ported into Python. Source of all the figures is Valmir P Moraes Filho.

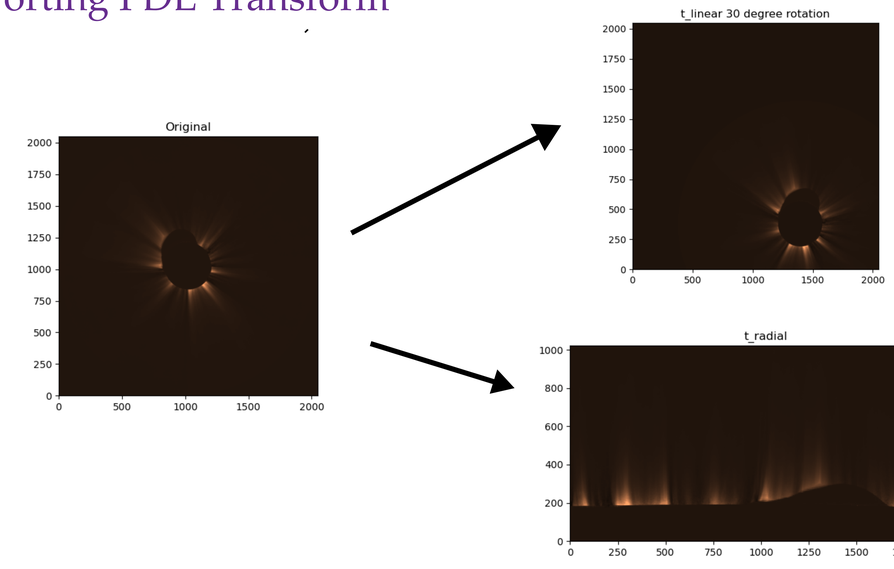
## Acknowledgements

This research was supported by the National Science Foundation REU Program, award #1659878. Funding was also provided by The National Center for Atmospheric Research funded by the NSF. I would also like to thank Valmir P Moraes Filho for his contribution of figures to this poster. I would also like to thank the whole PUNCH team, Marty, Claire, and everyone else for making this summer possible.

## 2) Perl Data Language (PDL) and PDL Transform

Perl Data Langauge (PDL) is an extremely powerful array-oriented numerical language. It was built as a free replacement for IDL and MatLab. However, PDL currently lacks support from the broader data science and heliophysics communities. This is mainly because of Python, which took over the as the mainstream programming language for this usage. PDL Transform is a convient way to represent coordinate transforms and resample images. There are two major methods- **Apply:** Transforms vectors into the new coordinate system; and **Map:** returns a remapped, resampled image on the new coordinate system.

## 3) Porting PDL Transform




**Left Figure)** Original COR2 image from the sun. This is at L2 data processing from the Deep Exposure Campaign.
**Top Right Figure)** T_Linear transform. This specific linear transform was a 30 degree rotation clockwise. T_linear can do rotations, arbitrary matrices, scaling factors, and pre/post shifting of the image.
**Bottom Right Figure)** T_radial transform, which converts Cartesian to radial/cylindrical coordinates. This allows for the unravelling of the sun that we see in the image.
PDL Transform is unlike anything that currently existed in Python, as it allows for a much more flexible coordinate transforms. PDL Transform is at the heart of porting over the Deep Exposure Campaign code.

## 4) Conclusion and Future Works

In Conclusion, PDL transform is mostly completed and we have layed the groundworks for porting over the rest of the Deep Exposure Campaign code into Python. PDL Transform has both working t_linear, t_radial, and t_compose. The biggest thing that needs to be impemented next is autoscaling in the Map function. This will allow for much easier t_radial transforms, as you don't have to do the scaling manually with t_linear.

The future is in the hands of the community. All of the code is available on my GitHub: github.com/Jake-R-W/pdl_tranforms. Feel free to improve the code as you see fit.