Draft version February 22, 2021

Typeset using LaTeX **twocolumn** style in AASTeX63

Radial Normalization of Solar Images

Chris R. Gilly[1,2] and Steven R. Cranmer[1,2]

[1] *Laboratory for Atmospheric and Space Physics*
[2] *Astrophysical and Planetary Sciences Department, University of Colorado Boulder*

ABSTRACT

The Solar Dynamics Observatory (SDO) has been in operation for a decade, and the images produced by its Atmospheric Imaging Assembly (AIA) are examined all across the globe, especially for on-disk diagnostics. But there is a great deal of information hiding in the low-signal upper regions of these images which appear to be dark. There have been many attempts to tease out this signal with varying success.

In this work we introduce a novel method to modify sun-centered solar images to have a constant dynamic range across the field of view. First, pixels are sorted by radial distance and the 75th and 2nd percentiles are found. These noisy curves are smoothed and fit with an polynomial function to provide a continuous scaling. Intensities are then normalized between zero and one, with outliers treated separately. This method produces visually striking, high contrast images of the Solar Corona out to the field of view of the instrument.

This reduction is being performed in perpetuity on an AWS t2.micro instance, with NRT imagery hosted at The Sun Now

The reduction engine is written in python and can be found on GitHub

1. INTRODUCTION AND MOTIVATION

Because the intensity of the solar corona drops off rapidly with heliocentric altitude, it is very difficult to take images of both the sun's disk and the overlaying corona with the same instrument. Historically, specialized devices have split the observation into multiple regions, with on-disk and off-limb observations being obtained by separate instruments.

As far back as [year], scientists have been recording the Sun's activity using telescopes. These early studies, carried out before even film cameras were invented, used hand-drawn sketches to keep track of solar activity (see, e.g. CITE and CITE). It wasn't until [year] that the first solar images were obtained on the ground. But ground-based observations have several limitations. First, the dynamic nature of the Earth's atmosphere requires pre- and post-processing of the images to remove

measurement artifacts (For example: speckle imaging, deformable mirrors, etc. (CITE and include more methods) Additionally, they are only capable of measuring the light which has not been absorbed by the Earth's atmosphere, excluding observation at many wavelengths. The only way around this limitation is to send an instrument into space.

Limited on-disk measurements were first taken by [spacecraft] in [year], providing [some kind of insight I can quote here]. Eventually the [SOHO?] mission was launched in [year], which provided the first synoptic images of the Sun in ultraviolet. Subsequently, the [Solar Dynamics Observatory (SDO)] mission took over in [2010], with its on-board Atmospheric Imaging Assembly (AIA) providing high resolution synoptic coverage of the UV Sun with a [90%] duty cycle. (CITE) There are many more solar imagers in flight at this time, such as [STEREO, PROBA, SUVI, etc] (CITE each one), but in this paper we will exclusively examine data from SDA/AIA.

While it is true that space borne instruments do not suffer from atmospheric effects like ground instruments, there are still many steps that must be taken to reduce the raw data (Level 0) into a science-quality image [Level 1.5?]. (See this AIA Reduction Paper for details (CITE)). The Joint Science Operations Center (JSOC) provides an archive of AIA images in the Level [1.5] category at (this link), (CITE), as well as Near Real Time (NRT) images with a 5-15 minute delay between the exposure and when it becomes available. These data can also be found at a number of mirrors: [The Sun Now, LASP LISIRD, solarflare.njit, SolarMOnitor, Virtual Solar Observatory, NOAA Dashboards, etc, etc]. Some of these mirrors also provide a wealth of context and analysis on the images, including Space Weather forecasting. To access the data programatically, one can use software like [Solar Soft for IDL, Sunpy for Python, etc].

Figure 2(b) shows the reduction of the SDO images to a standard Level 1.5 [maybe also include a comparison with a completely raw image?]. Notice that while the on-disk regions are visible, there is very little imagery off the limb. If one is interested in seeing the details in the solar corona, an additional step must be taken to bring out the contrast across the whole image. There are many methods in the literature that can be used: taking the square root or logarithm of the entire array (cite?), multiplying each pixel by its radial value r^{NUM} (cite?), Multi-scale Gaussian Normalization (**?**), Noise Gating(**?**), and the aia_rfilter in SSW written by Patrick McCauley (CITE), to name a few. These methods come with some limitations: *Why do the old methods need to be improved upon? [Not high enough contrast, too slow, etc. Using multiple methods concurrently - does it help?]*

In this work, we introduce a novel algorithm called SUNBACK (*Should probably rename the core algorithm vs the background updator*) which normalizes images of the Sun's corona using the percentile function. Section 2 details the new algorithm. Section 3 then compares and contrasts our results with those obtained using other methods from the literature. Finally, Section 4 provides a discussion and conclusion.

2. METHOD

This section describes the SUNBACK algorithm as implemented in python, hosted on GitHub. (see Appendix A for usage instructions)

The reduction pipeline operates as follows (simplified):

1. An image I (either a fits file, file path, or a np.ndarray) is passed into the Modify class by a user.

2. Load the image I. Add $\min(I)$ to I, setting the minimum intensity to 0.

3. Compute a radius array R with the same shape as I so that each pixel has a known $r = \sqrt{x^2 + y^2}$. (This is probably what one would change if one wanted to work on off-limb data).

4. Flatten I and bin the intensities by radius. This produces a distribution of intensity values at each bin height. Figure 1(a) shows these intensity values as black points vs radius.

5. Compute an upper and lower percentile of the intensities in each bin. Figure 1(a) displays our choice of the 75th percentile P_{75} and the 2nd percentile P_{02} as colored curves.

6. Use a Savgol filter to smooth the percentile curves, treating the regions independently.

7. Fit a polynomial to the on-disk curve to help mitigate over-bright regions.

8. Concatenate the regions and Savgol smooth them again.

9. Normalize the input image using these curves and the standard formula

$$\frac{x - \min(x)}{\max(x) - \min(x)} \tag{1}$$

10. Handle outliers like NANs, hot, and cold pixels

11. Vignette the image, truncating the image at a given radius

12. Save the image to disk, then exit.

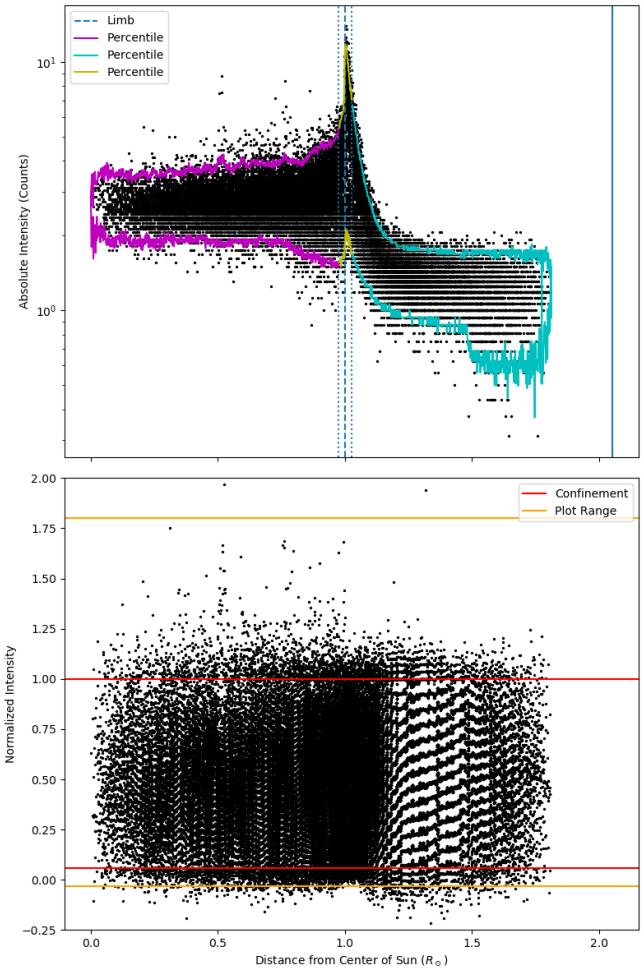


Figure 1: Pixel Intensity vs Radius

Figure 1 (a) shows the intensity values at each height in black, along with the 75th and 2nd percentile curves.

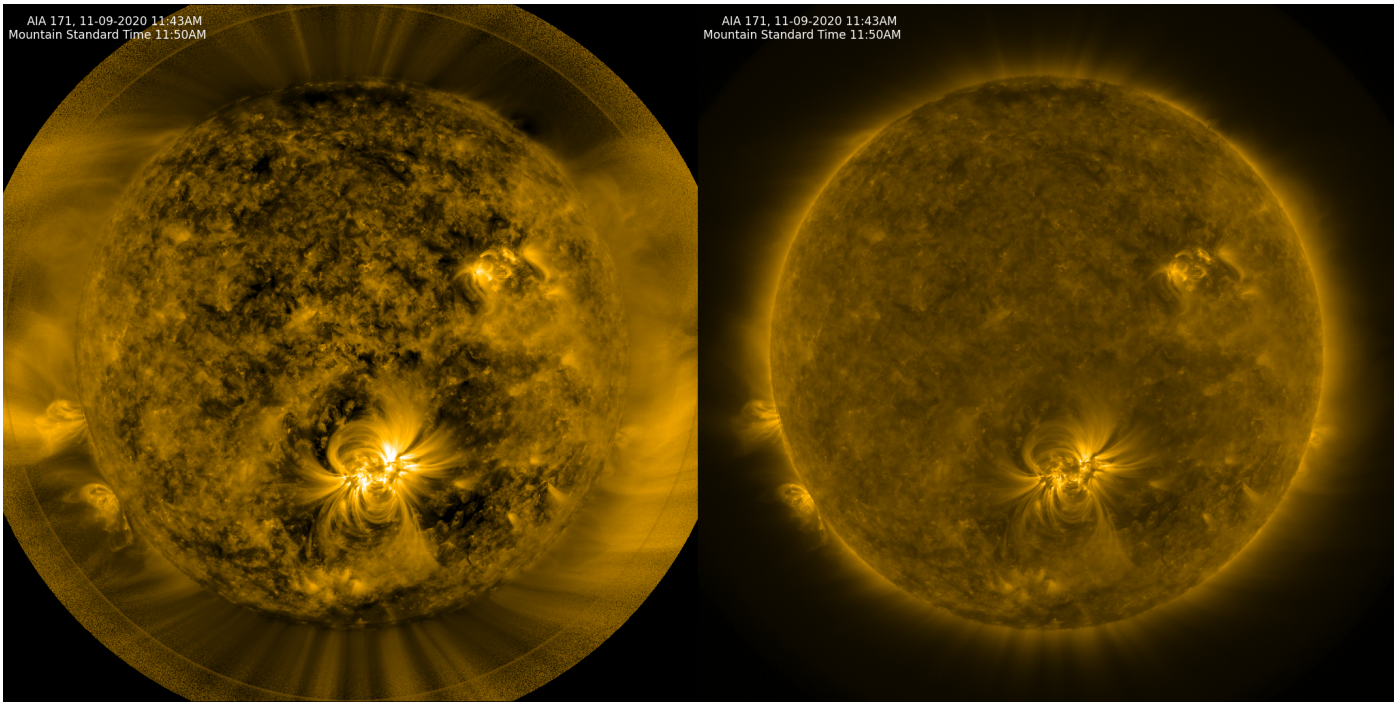


Figure 2: Normalized Image on Left, Original on Right

Figure 1 (b) shows the result after normalizing the intensities to these percentile curves.

Figure 2 compares a processed image beside the original.

3. COMPARISON WITH METHODS FROM THE LITERATURE

There are a variety of ways to reduce a solar image of this type. In this section we compare the results of these different algorithms and examine the benefits of each. Figure 3 shows these images side by side.

3.1. *Isotropic Scaling*

Applying the same function to all pixels. This includes things like taking the root or the logarithm of each pixel.

3.2. *Radial Scaling*

Applying a coordinate-based scaling. This typically means multiplying each pixel by a power of its heliocentric distance r.

3.3. *AIA_rfilt*

An algorithm written by Patrick McCauley, in SSW. (**?**)

3.4. *Multi-Scale Gaussian Normalization*

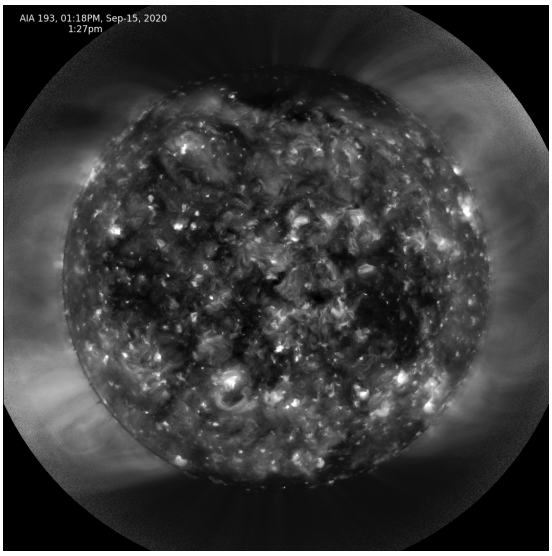
Applying multiple distinct scalings (**?**)

3.5. *Noise Gating*

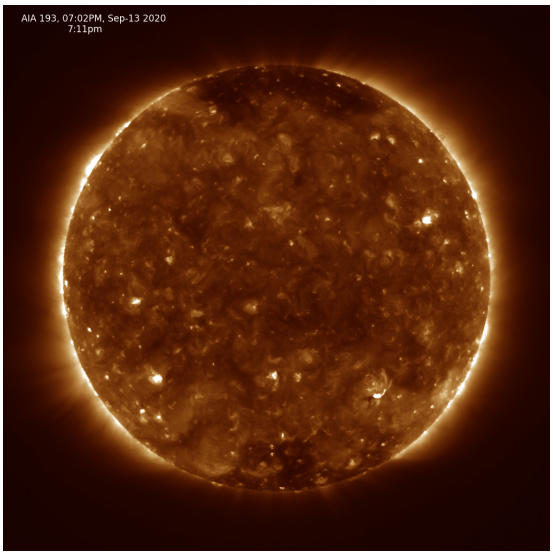
Fourier Filtering to Truncate Noise (**?**)

4. DISCUSSION + CONCLUSIONS

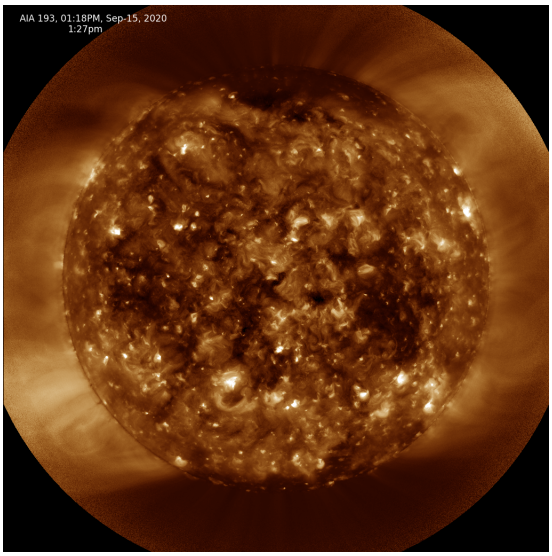
There will probably be some.



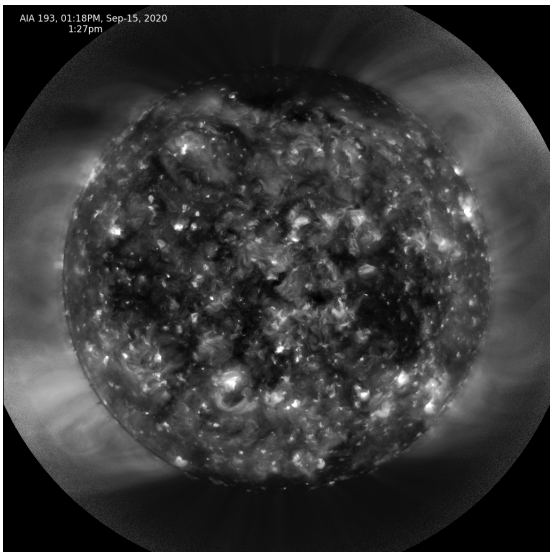
(a) Raw Level 0 Image



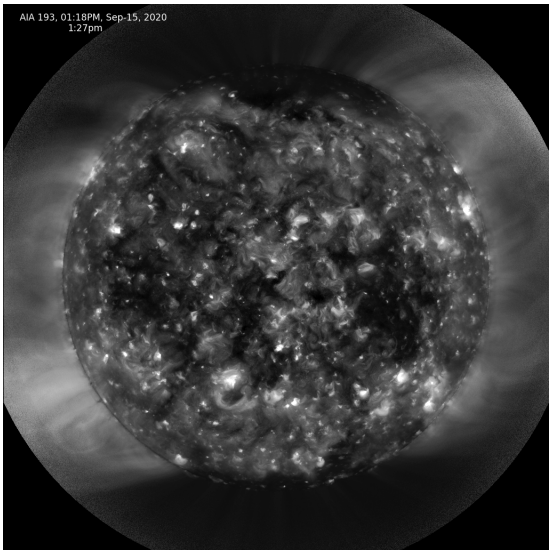
(b) Reduced to Level 1.5 Image



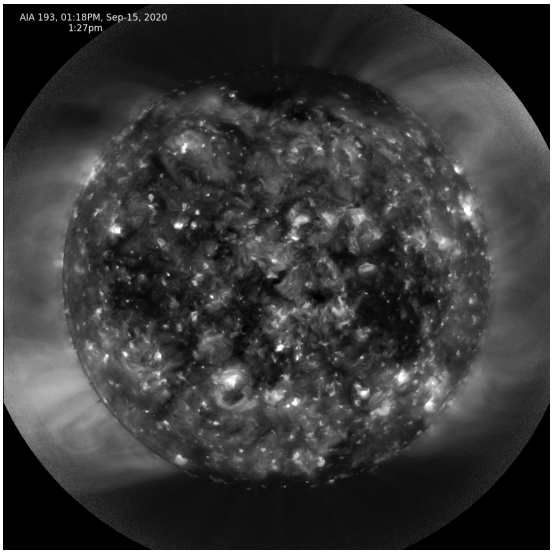
(c) Reduced with Sunback



(d) Reduced with MSGN



(e) Reduced with another method



(f) Reduced with another method

Figure 3: Image of the sun shown with various reductions applied. Images in BW are placeholders.

5. DISSEMINATION + FUTURE WORK

I have many ideas for ways that radial filtering can be used for both outreach and research.

5.1. *Outreach*

When I showed AIA images to third-graders in 2017, they didn't believe it was even the Sun at first. Attempting to introduce realistic solar images to elementary schoolers could perhaps inspire them to join our Helio community.

Website - I am currently hosting only NRT images on my webpage at www.gilly.space/sun. I would love to also include a video of the last 48 hours, like the ones seen at https://www.thesuntoday.org/the-sun-now/.

Planetarium - having a "sun pizza" on the dome as people filter into the theater would be very cool. This video could also be hosted on my website and served to an arbitrary number of clients.

5.2. *Research*

I'd like to get my filter incorporated into the FORWARD code (cite).

I'd like to incorporate the filter into the SWx-TREC Model Staging Platform + Data Portal (Cite)

I'd like to incorporate the filter into Sunpy (CITE)

I'd like to make the core algorithm available on PyPi

5.3. *Future Work*

There is a review paper about solar coronagraphy in preparation by (Naty Alzate) that I'd like to collaborate on.

5.4. *Versions*

sunback_filt is the core algorithm with no bells or whistles

sunback_aia is the wrapper which downloads the raw data, performs the algorithm, then hosts on my website by uploading to AWS.

sunback is the package on pypi which downloads the images and sets your desktop background

ACKNOWLEDGMENTS

The authors gratefully acknowledge XX, YY, and ZZ for many valuable discussions. This work was supported by the National Aeronautics and Space Administration (NASA) under grants NNX15AW33G and NNX16AG87G, and by the National Science Foundation (NSF) under grants 1540094 and 1613207. Thanks also to Juri Toomre's Laboratory for Computational Dynamics for providing us with computing time.

Software: NASA's Astrophysics Data System (ADS), Python version 3.6.5 (**?**), numpy (**?**), scipy (**?**), scikit-image (**?**), MPI4py (**?**), matplotlib (**?**), and CHIANTI v8 (**?**)

APPENDIX

A. GETTING STARTED

To get a local copy up and running follow these simple example steps.
Prerequisites: You need python 3, and Conda is probably your best bet.

A.1. *Installation*

1. Clone the repo

```
git clone https://github.com/GillySpace27/sunback_engine.git
```

2. Install required packages

```
pip install -r requirements.txt
```

3. Confirm it works (tests will run)

```
python sunback_filt.py
```

A.2. *Usage Examples*

To perform the reduction using a fits file, you can provide the path to the file to the Modify Class:

```
 from sunback_filt import Modify
input_path = "/data/171_MR.fits"
reduced = Modify(input_path)
```

One can also perform the reduction directly on any numpy NDArray:

```
from sunback_filt import Modify, load_file
input_path = "/data/171_MR.fits"
input_file = load_file(input_path)
```

```
reduced = Modify(input_file)
```

Defaults for Modify (Top Level Class):

```
Modify(data=None, image_data=None, orig=False, show=False, verb=False)
```

with full_name, save_path, time_string, shape = self.image_data