**LEISA Post-pipeline De-fringing Algorithm Description Document**
**V1.0: Amy Simon and Matthew Montanaro, May 9, 2024**

To Use

Fringe flats are identical in size to the Level 2 calibrated science data. Divide each frame of the Level 2 data cubes by the corresponding fringe flat (fflat) to produce the de-fringed, calibrated, data. Further details on the fringing and removal method are described below for the interested user.

Introduction

LEISA data can show optical fringes that vary with location on the detector (Figure 1). These features are prevalent at low illumination levels but can occur in any data set. The fringes are not accurately described by a single mathematical function and vary by filter, location on the detector, and possibly instrument temperature. In well-behaved data these can be removed in a variety of methods. Here we describe the most consistent removal method, as developed and tested by the L'Ralph team and used in the fringe flats that have been delivered to the PDS. It should be noted that the flats will not remove all fringes but are the best compromise between fringe removal and maintaining real pixel to pixel variation.

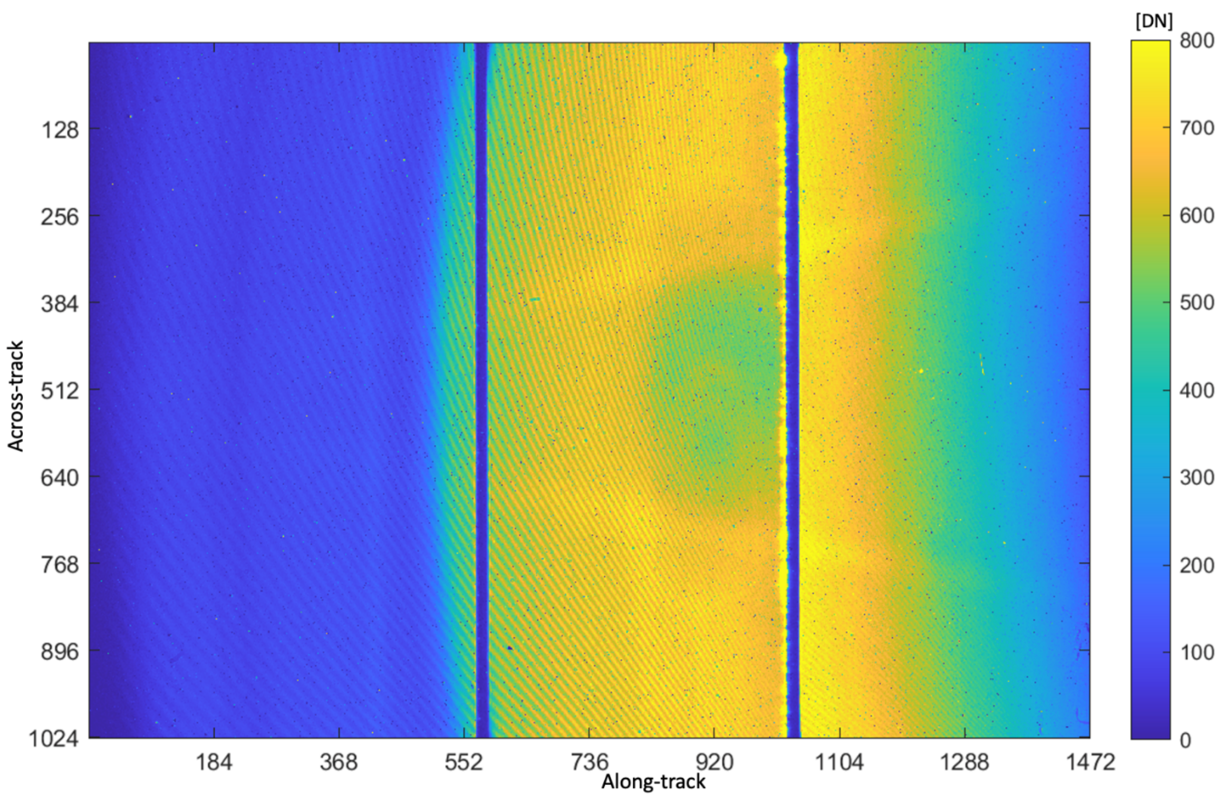


**Figure 1.** Example fringing on in-flight internal Filament data (dark subtracted). Fringes can exceed 10% of the signal in some cases.

**ALGORITHM DESCRIPTION:**

A flat-field image is derived from LEISA data itself by fitting a 1-D Gaussian model to the oscillating pattern on a per-row basis and dividing out a wide-area smoothed version of the fitted Gaussian model values. The intent is to isolate the fringing pattern as best as possible without over-smoothing pixel-to-pixel information.

**INPUT:**

Single frame of LEISA imagery. It is assumed that there are 1472 columns (along-track) and more than one row (across-track). The image data may be in units of counts or processed radiance and can be unsigned integers or floating-point values. The bad pixel map should already have been applied prior to the de-fringing procedure.

**OUTPUT:**

Double-precision floating point array the same size as the input array with decimal values varying about one representing the flat-field value for each pixel in the image.

**PROCEDURE:**

Assumption is the input data array has been corrected for bad pixels utilizing the LEISA bad pixel map.

**Step 1 (pre-conditioning step):**

The input data array is converted to a double-precision floating-point data type. To condition the image data for de-fringing, single-pixel artifacts need to be removed that may still remain after the bad pixel map has been applied (see Figure 2). A 2-D median filter is applied to the image data to accomplish this. A 5x5 filter is applied to LVF3 and a 3x3 filter is applied to LVF2 and LVF1. These sizes were chosen by trial and error. Pixels with a value of zero or NaN are replaced with an interpolated value from surrounding pixels. (see Figure 3 for an example of the median filtering result).

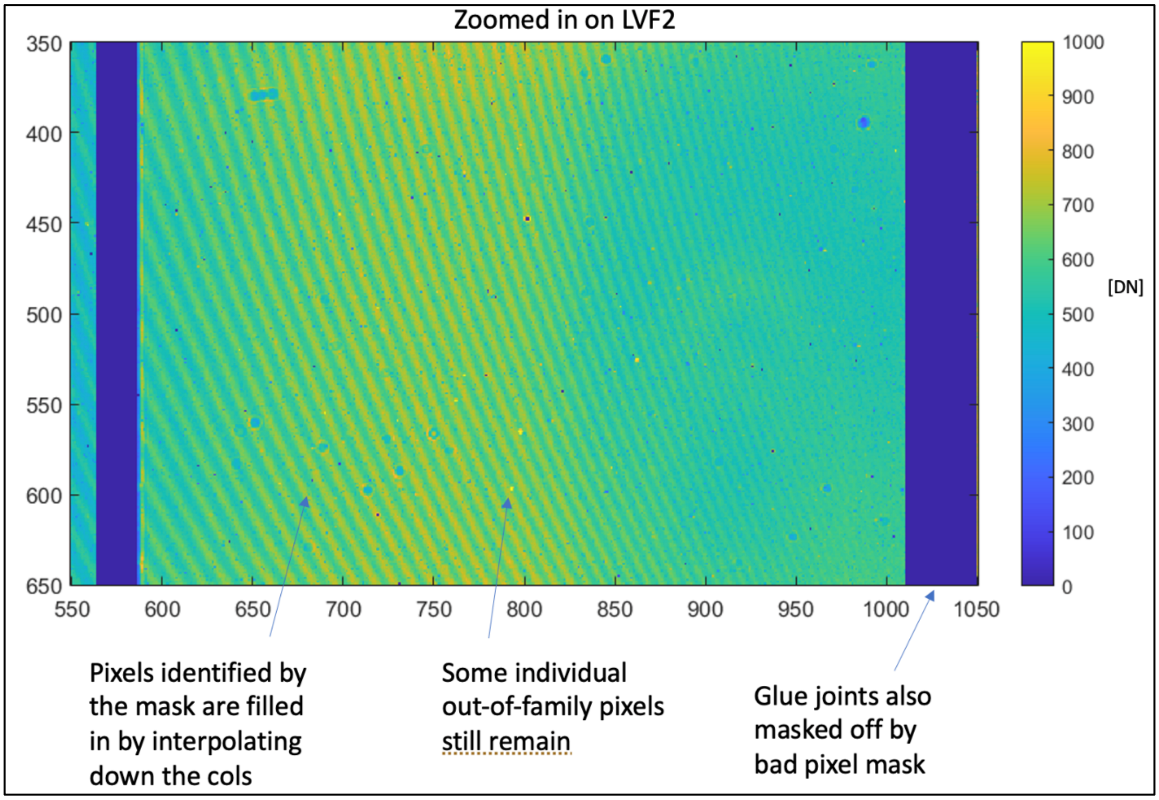


**Figure 2.** Applying the bad pixel correction to a zoomed portion of Figure 1.

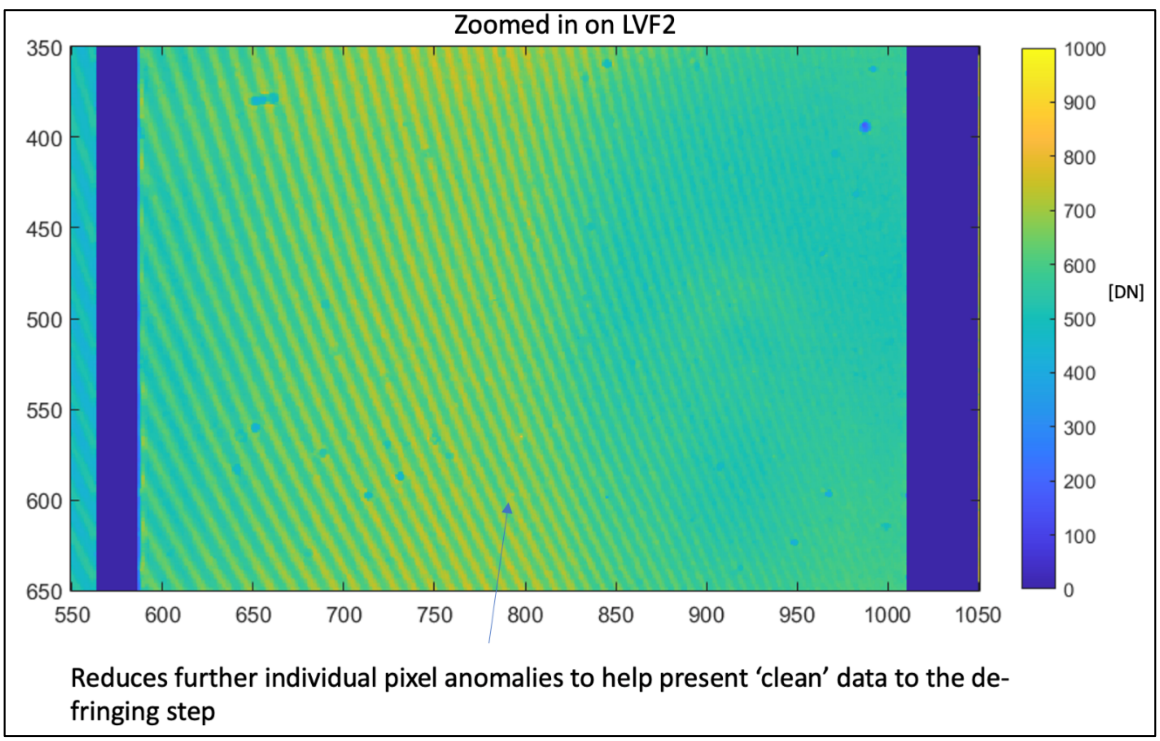


**Figure 3.** Data from Figure 2 after median smoothing

**Step 2 (flat-field generation):**

Once the image data has been conditioned with the median filter, the de-fringing flat is derived on a per-row basis.

**Step 2a**: For a given row, a 1-D Gaussian function is fit to a 7-element sliding window. At either end of the row, the window is truncated as needed. The result is multiple (up to 7) Gaussian values per column representing the value of the fitted Gaussian function for each overlapping sliding window (See Figure 4). The modeled value for a particular column is taken as the mean of the three Gaussian values when the sliding window is centered on that column and centered one column to the right and to the left of that column. This is derived for every column in the row and yields a single modeled value per column.

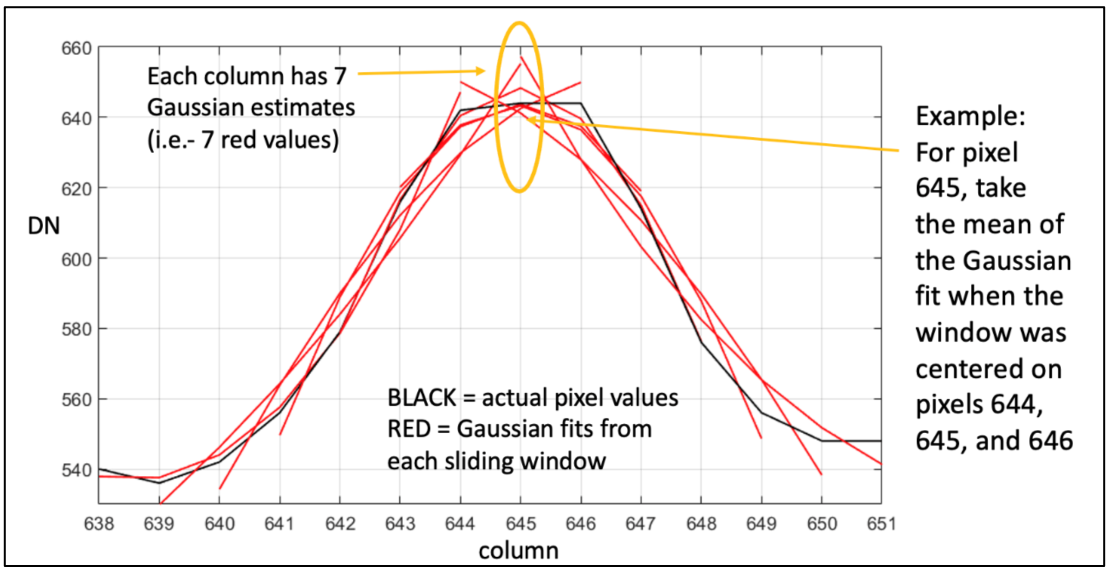


**Figure 4.** Sliding Gaussian fits to find the mean for a particular column.

**Step 2b**: To isolate only the oscillating pattern, a local average needs to be derived that captures the local, non-oscillating signal level. This is performed by smoothing the modeled values from the previous step by fitting a quadratic function to a wide sliding window across the row. The wide-area sliding window encompasses several wavelengths of oscillation to derive a good mean value. Window sizes of 31 elements, 21 elements, and 13 elements are used for LVF3, LVF2, and LVF1, respectively. A second stage of smoothing is then applied using a 41 element window across the entire row. The window sizes were decided by trial and error.

**Step 2c**: Finally, the flat-field values for the row are derived by dividing the Gaussian model values by the smoothed model values.

These steps (2a through 2c) are repeated for every row to yield a flat-field image the same size as the input array (see Figure 5 for an example). The user would divide the flat image into the input image to reduce the fringing pattern (see Figure 6 for an example application).

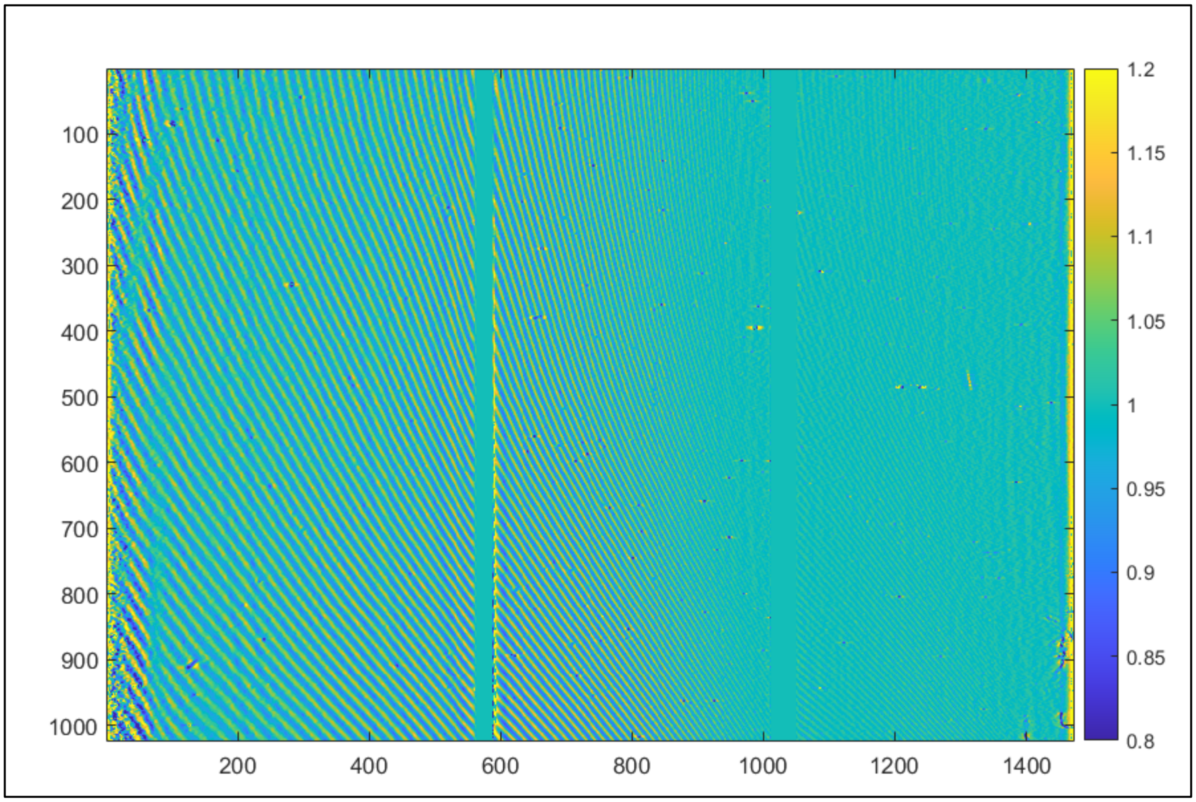


**Figure 5.** Example of the derived Fringe Flat using the image in Figure 1 as the input.

<u>Caveats</u>

For areas of an image without good illumination, this process might produce noisy flats that can be masked out if applying it would produce unacceptable artifacts.

This process was applied to Dinkinesh datasets by first combining all frames into a single frame representing the maximum value per pixel over all frames. The flat is then derived from this composite array.

This process is designed for extended illumination targets and has not been attempted on point-source data.

Example application:

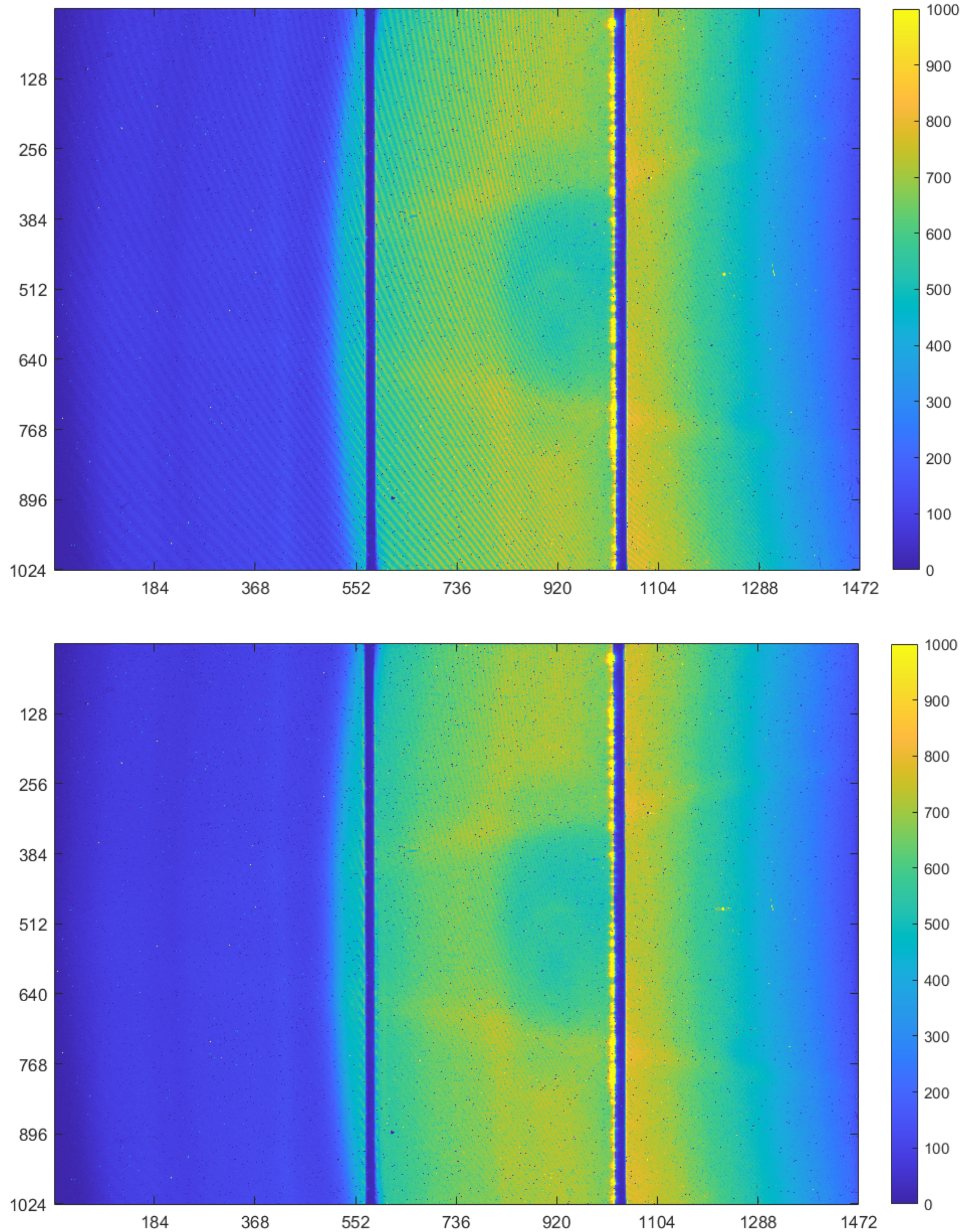


**Figure 6.** Data before (top) and after (bottom) applying the flat.