LEISA Post-pipeline De-fringing Algorithm Description Document
V1.0: Amy Simon and Matthew Montanaro, May 9, 2024
V1.1: Amy Simon and Matthew Montanaro, December 12, 2024
V1.2: Matthew Montanaro, July 18, 2025

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. Note that these files may be referred to as "flat field" files in the PDS4 xml labels due to limitations in the available label values in the PDS4 information model. Any "flat" file in the context of the LEISA instrument is a fringe flat file.

Introduction

LEISA image frames can show a cyclic, optical fringing pattern that varies with location on the detector (see 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. The methodology described here has yielded the most consistent results to significantly reduce the fringing artifacts in LEISA imagery. This method of developing a flat-field image that specifically targets the fringing pattern was developed and tested by the L'Ralph instrument 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 fringing artifacts but are the best compromise between fringe removal and maintaining real pixel to pixel variation.

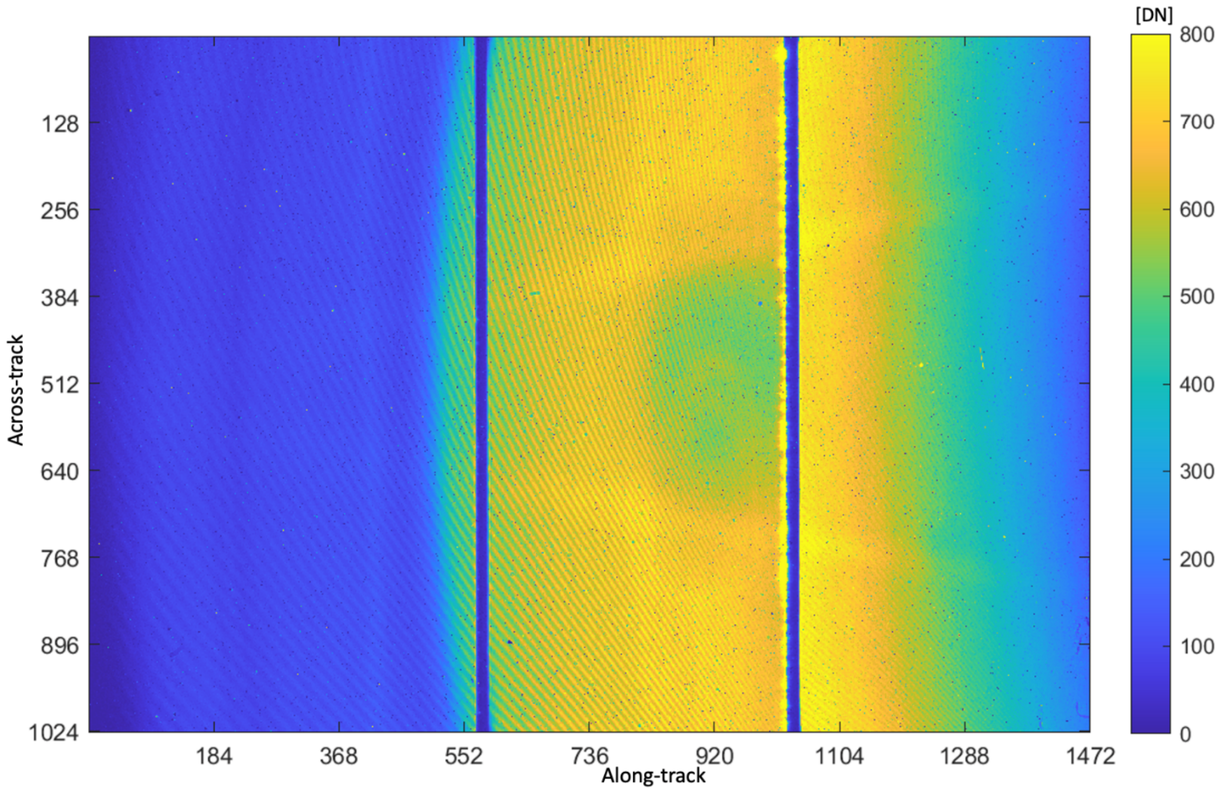

Figure 1. Example fringing on in-flight internal Filament data (dark subtracted). Fringes can exceed 10% of the signal in some cases.

ALGORITHM DESCRIPTION:

A flat-field image targeting the fringe pattern 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. The number of columns (along-track dimension) can be any value greater than seven and the number of rows (across-track dimension) must be greater than one.

The user specifies the column window as a two-element array of [start channel, number of channels]. These values can be obtained from the FITS meta data field of "LEISA along track start channel" and "LEISA along track number of channels", respectively.

The user also specifies whether the LEISA image is in super-pixel mode. The FITS meta data field "playback header type" will identify either super pixel mode ("LEISA_SUPER") or normal mode

("LEISA_CDS"). For super pixel mode, the algorithm will adjust the Gaussian fit window to an optimal value vs. normal mode.

The input 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. Flat-field values for columns under the glue bond areas are set to one.

PROCEDURE:

Assumption is the input data array has been corrected for bad pixels utilizing the LEISA bad pixel map. The input data array is converted to a double-precision floating-point data type. Given the user-specified column window size and super pixel mode, internal algorithm parameters are set to the optimal values for the Gaussian window and each LVF section of the array.

Step 1 (pre-conditioning step):

To condition the image data for de-fringing, single-pixel artifacts need to be removed that may 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. Figure 3 is an example result of the median filtering.

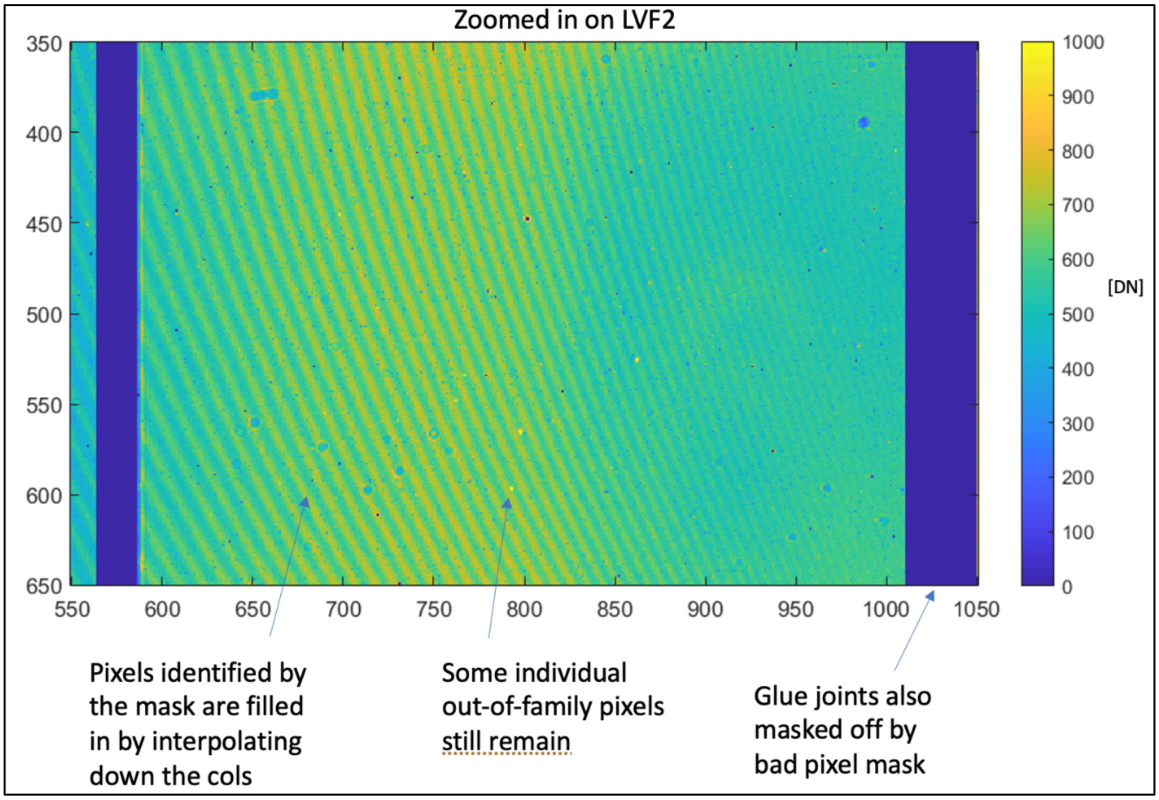

Figure 2. Applying the bad pixel correction to a zoomed portion of Figure 1.

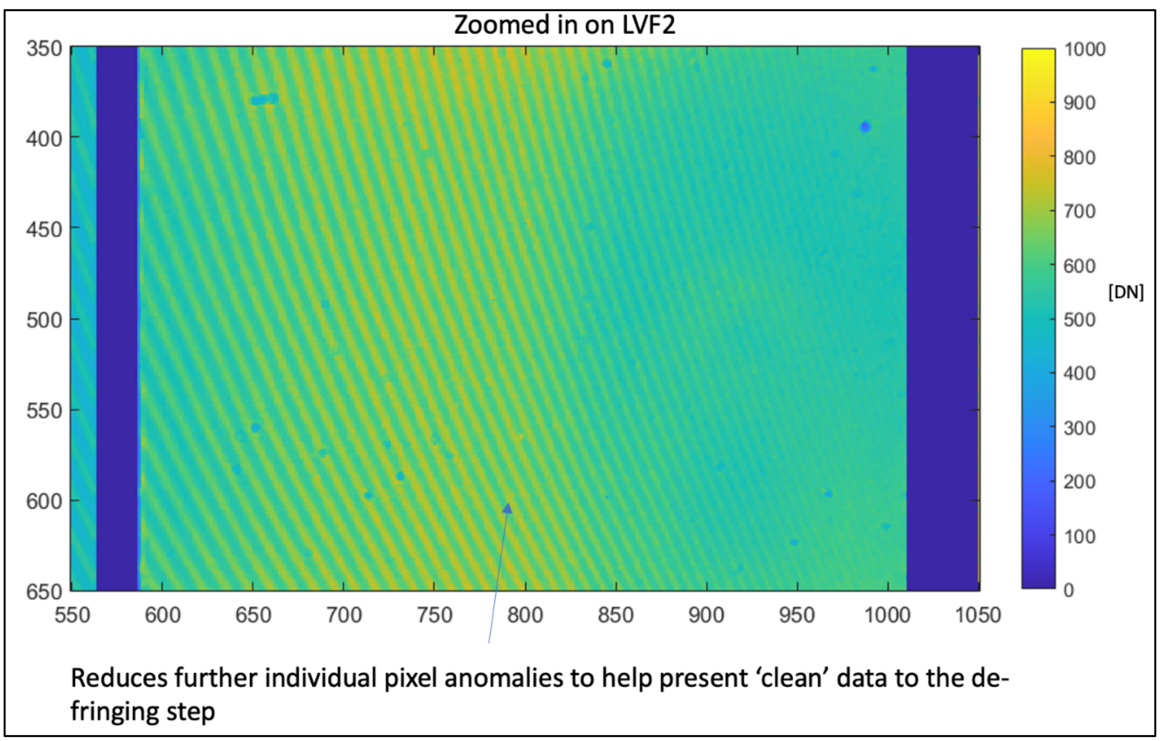

Figure 3. Data from Figure 2 after median smoothing

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 (or a 3-element sliding window for super-pixel data). The window widths were determined through trial and error. At either end of the row, the window is truncated as needed. The result is multiple 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.

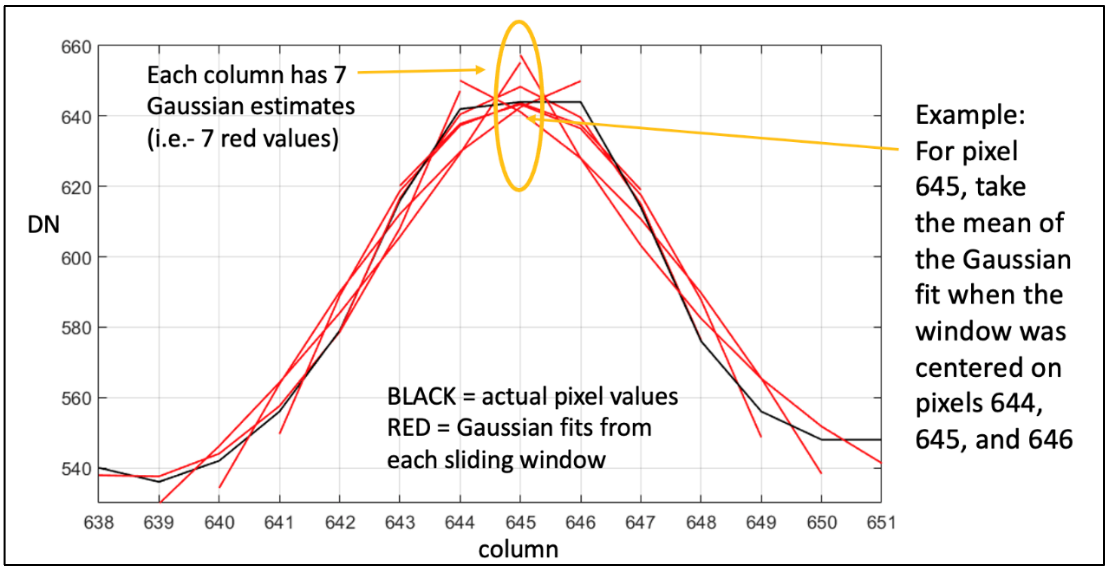

Figure 4. Sliding Gaussian fits to find the mean for a particular column.

Step 2b: To isolate only the oscillating pattern, a wide-area 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). Pixels under the glue bond joints have a flat-field value of one. Flat-field values above 1.3 or below 0.7 are assigned a value of one to limit extreme values in the output array. The user may alter these thresholds in the source code if desired.

To apply the flat, the user would divide the flat image into the desired fringe image to reduce the fringing pattern (see Figure 6 for an example application).

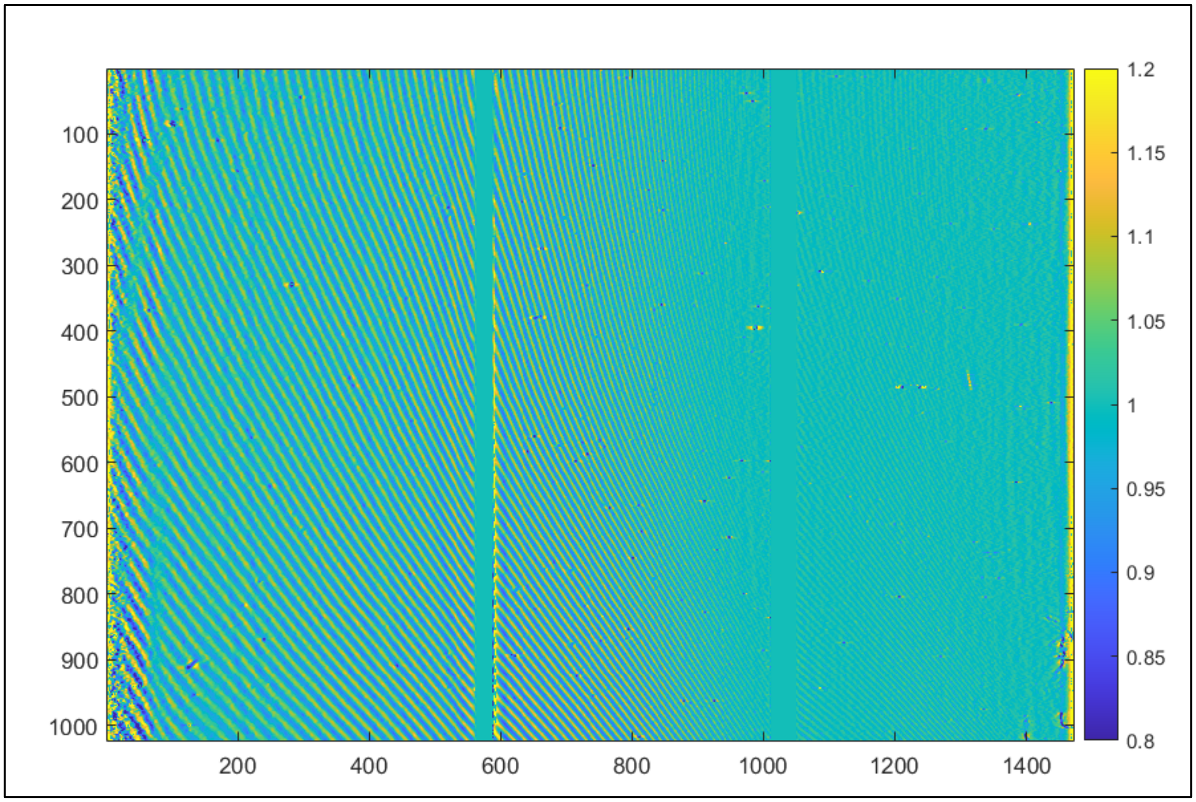

Figure 5. Example of the derived Fringe Flat using the image in Figure 1 as the input.

Example application:

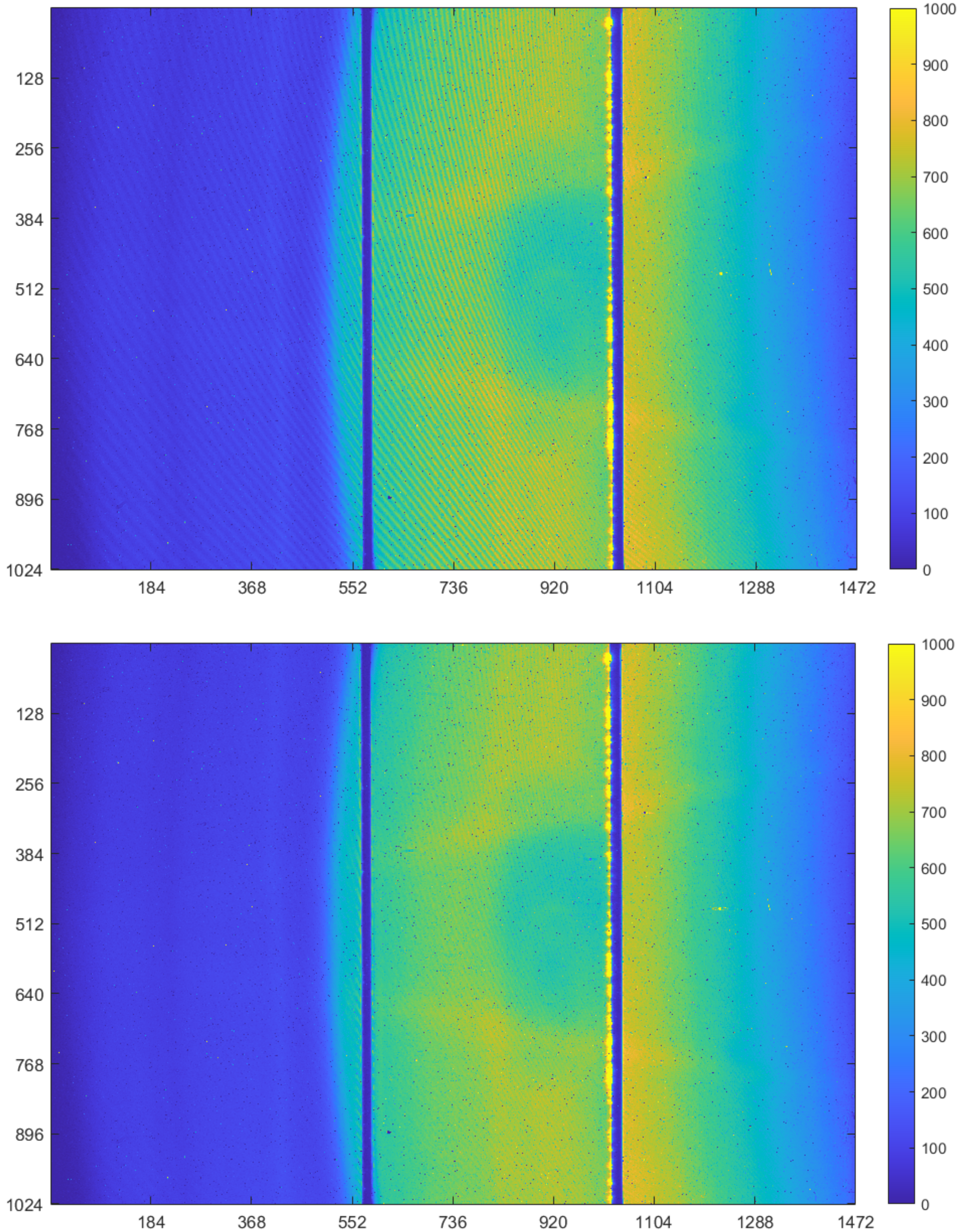

Figure 6. LEISA image data before (top) and after (bottom) applying the derived flat.

Caveats and Known Issues

- 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 is designed for extended illumination targets and has not been thoroughly tested on point-source data (e.g., stellar data).
- For scanned datasets with a high number of frames (e.g., Dinkinesh), the fringing pattern may only be evident over detectors illuminated by the target. In this case, the maximum value of each detector across all frames can be calculated which will make the fringing pattern obvious (see Figure 7). This maximum value array can then be used as input to the de-fringing function and the output flat applied to each frame of the scan.

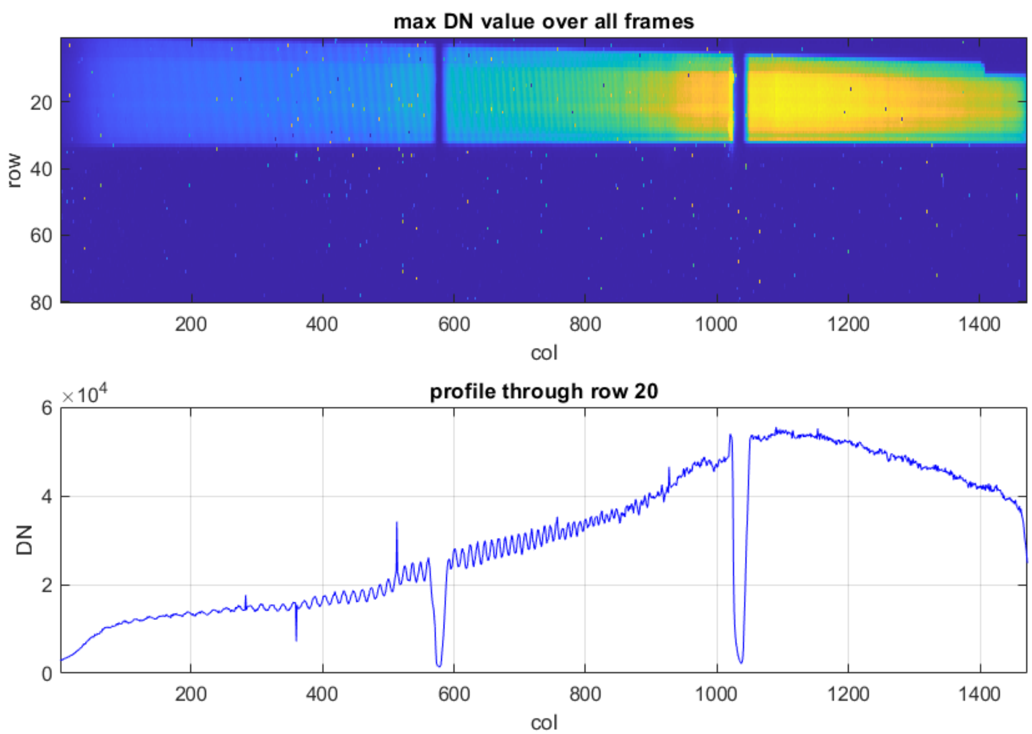

Figure 7. Input fringe image constructed for a Dinkinesh scan by calculating the maximum value of each detector over all 1022 frames (top). A profile through the maximum value image illustrates the fringing pattern across the columns (bottom). [lei_0752129712_02300_eng_01.fit]