

# Cosmic ray removal in Demetra

Kevin Gurney

May 28, 2021

## 1 Background

Cosmic ray removal is available in some packages but can give rise to artifacts, especially if used with binned images that give vertically narrow spectra. I investigated the removal of cosmic rays defects in astronomical images and there appears to be an ‘industry standard’ method in the Laplacian Cosmic Ray Identification algorithm of PG Dokkum [2]. It is used in the standard IRAF library and has a Python implementation - [1]

## 2 Method

We don’t detail the LAcosmic algorithm here. Rather we describe how we have deployed it in the Demetra environment. The algorithm works best (apparently) on images which have been processed with bias, darks and flats. These are available in the `Object_P` files in Demetra after the running the Preprocess step. It is at this stage that we run the cosmic ray removal. The Python code we have written is best run from a .bat file in the sub-directory containing the `_P` files. This short file passes a parameter which gives a detection threshold for the algorithm to work with (see below). The algorithm will also pick up any hot pixels left over from previous pre-processing

The code searches for `_P` files, and copies them to a series of backup files prefixed `BAK_` before processing. It then replaces the originals with the new ones with cosmic rays removed. In addition, it produces a series of map files prefixed `map_` which show where the cosmic rays were identified. Finally it also shows an aggregated map superimposing all the individual maps in `full_map.fits`. The individual maps have binary-valued pixels (‘cosmic ray represent or not’) whereas the overall map has integral values, so that hot pixels present across many subs will have a value equal to the number of instances therein.

The algorithm can be compute intensive with increasing time for larger images. Therefore, it is useful to crop the original spectra subs before doing anything in Demetra. This is also useful for simply saving file space in genera. Use a batch crop utility (I use the one in Hyperion Prism) for doing this.

Having run the cosmic ray removal, proceed in Demetra with the next step (‘Geometry’) in the normal way.

### 2.1 Choosing a threshold

Example parameters given in online resources tend to be good for deep sky images which have a lot of signal compared to the (mainly dark) spectra we are processing. We have found that for 300s exposures on fainter spectra a good parameter is around 15. For bright spectra a value of around 85 seems to work well. If the parameter is set for a more aggressive approach, many of the bright pixels in the extreme left of the frame (left over from the application of flats) are dealt with. While this is not necessarily a bad thing, it may be a sign that artifacts could be about to emerge in this area of the image (although I have never seen any as such).

It may be useful when exploring this algorithm for your own spectra to look for any corruption of spectrum pixels in an auxiliary program. I have found an image blink utility (I use that in Prism) to

be useful in this regard. Thus, load up the full map and ‘blink compare’ it with the spectrum. So far I have not seen any corruption of spectra over a wide parameter range.

## References

- [1] “lacosmic - lacosmic v0.1.dev65+g88a916b.” [Online]. Available: <https://lacosmic.readthedocs.io/en/latest/index.html>
- [2] P. G. van Dokkum, “Cosmic-Ray Rejection by Laplacian Edge Detection,” *Publications of the Astronomical Society of the Pacific*, vol. 113, pp. 1420–1427, Nov. 2001. [Online]. Available: <http://adsabs.harvard.edu/abs/2001PASP..113.1420V>