Cosmic-Ray detection and masking

Algorithm by Pieter van Dokkum
Implemented and modified by Marino Maiorino & Santiago Serrano

Objective

Identify and mask cosmic-rays present in astronomical images for the data reduction process.

The Algorithm

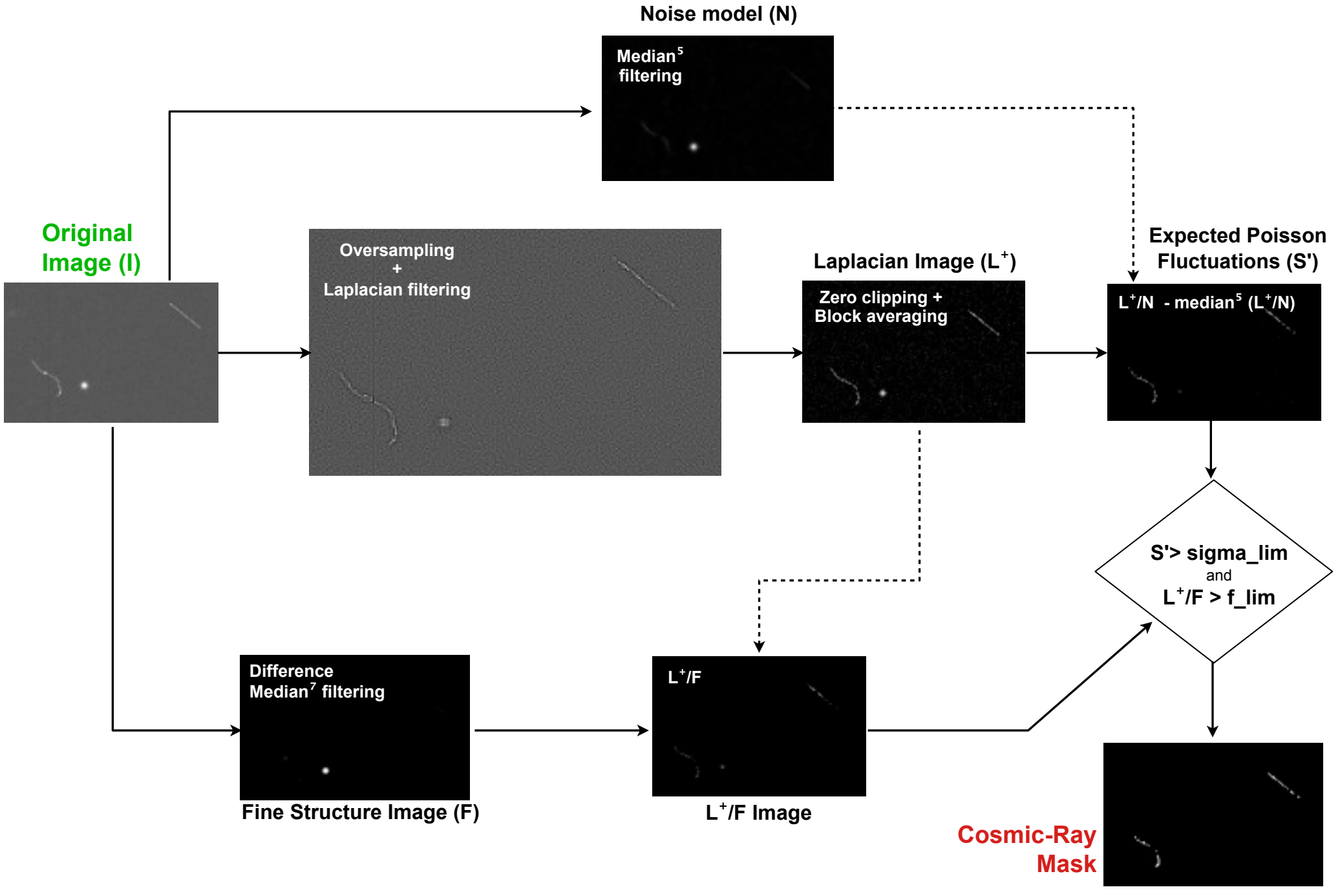
The concept was designed by Pieter Van Dokkum. The method is described in his paper: arXiv:astro-ph/0108003v1 [2001]

The code

The code is written in Python and requires some analysis libraries:

- pyfits
- numpy
- imaging
- scipy

Algorithm Diagram



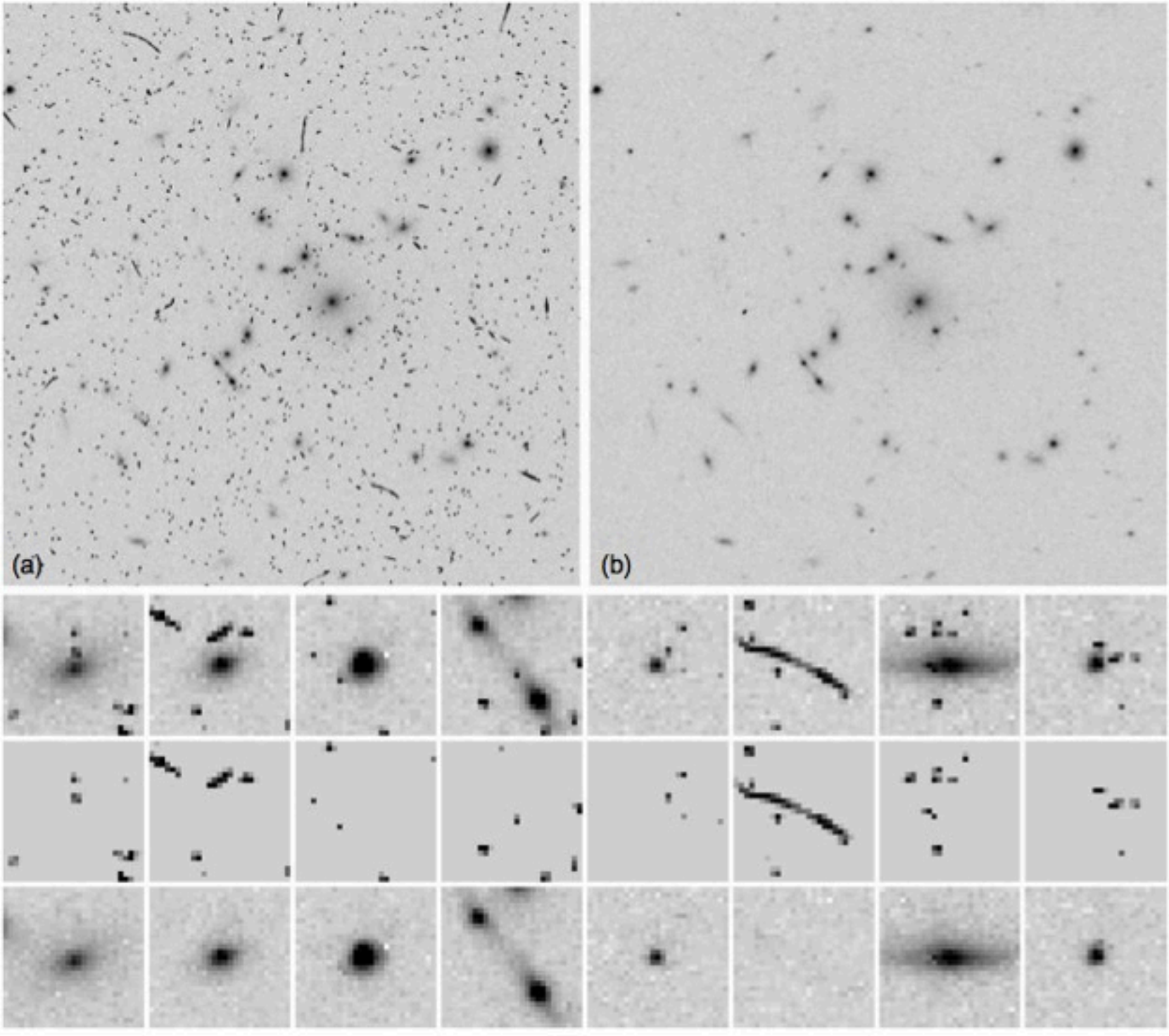
Input Parameters

The current code runs with the following parameters that allow different results:

```
#Algorithm Parameters
    iterations          #Number of laplacian filtering passes
    subsampling         #Subsampling for filtering (should not affect to result)
    auto_limits         #f_lim & sigma_lim are adapted to the image or fix
    f_lim               #Usually 0.05  - Ignored in auto limits mode
    sigma_lim           #Usually 1     - Ignored in auto limits mode
    f_lim_tolerance_multiplier      #Flim tolerance
    sigma_lim_tolerance_multiplier  #Sigma tolerance
    F_image_mode        #Pieter van Dokkum original mode or Time optimized mode
    recon_type          #Image Reconstruction type.
                        #Valid any surrounding pixels  2:Valid axis only
```

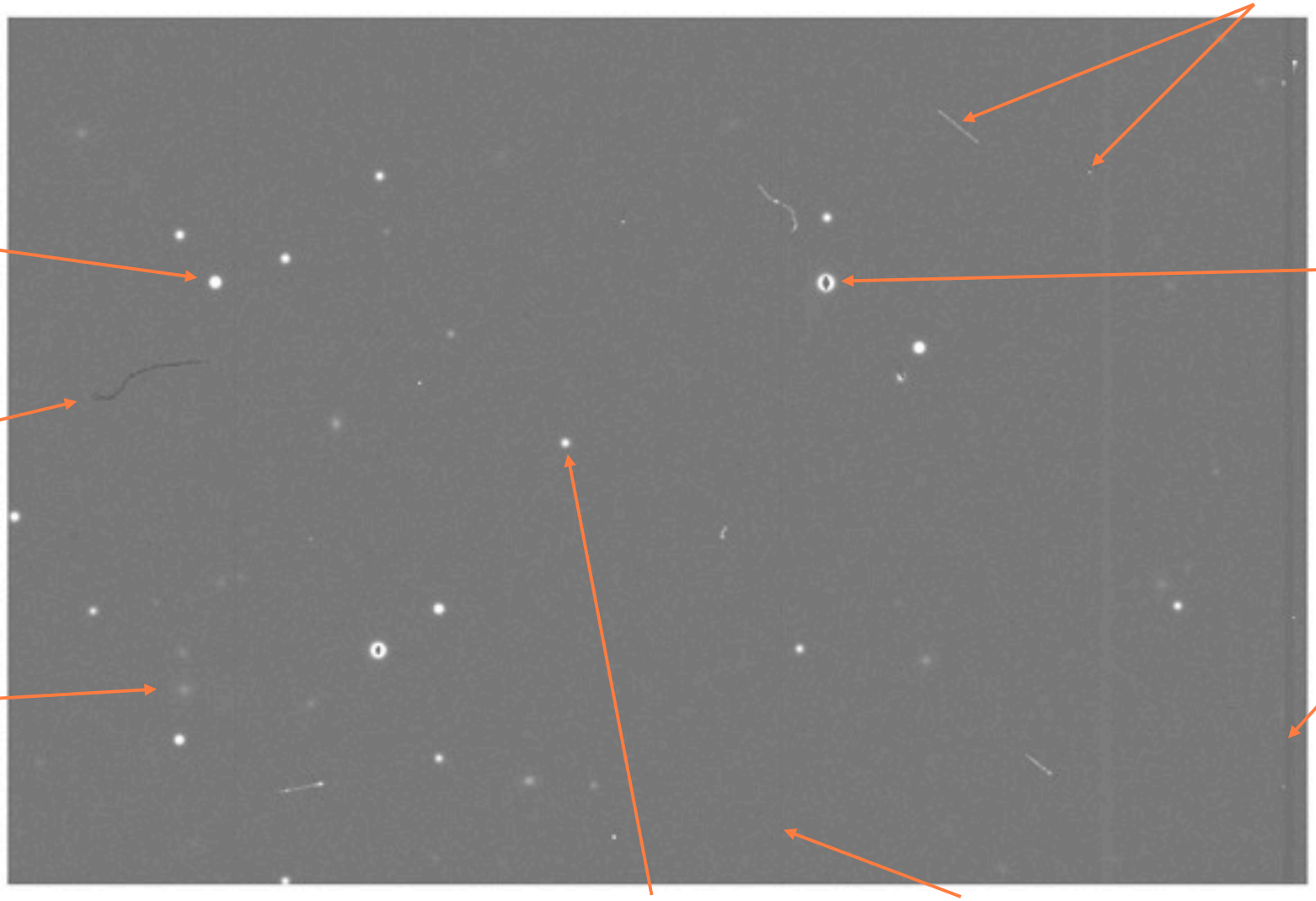
Results

From author's analysis in a HST WFPC2 image



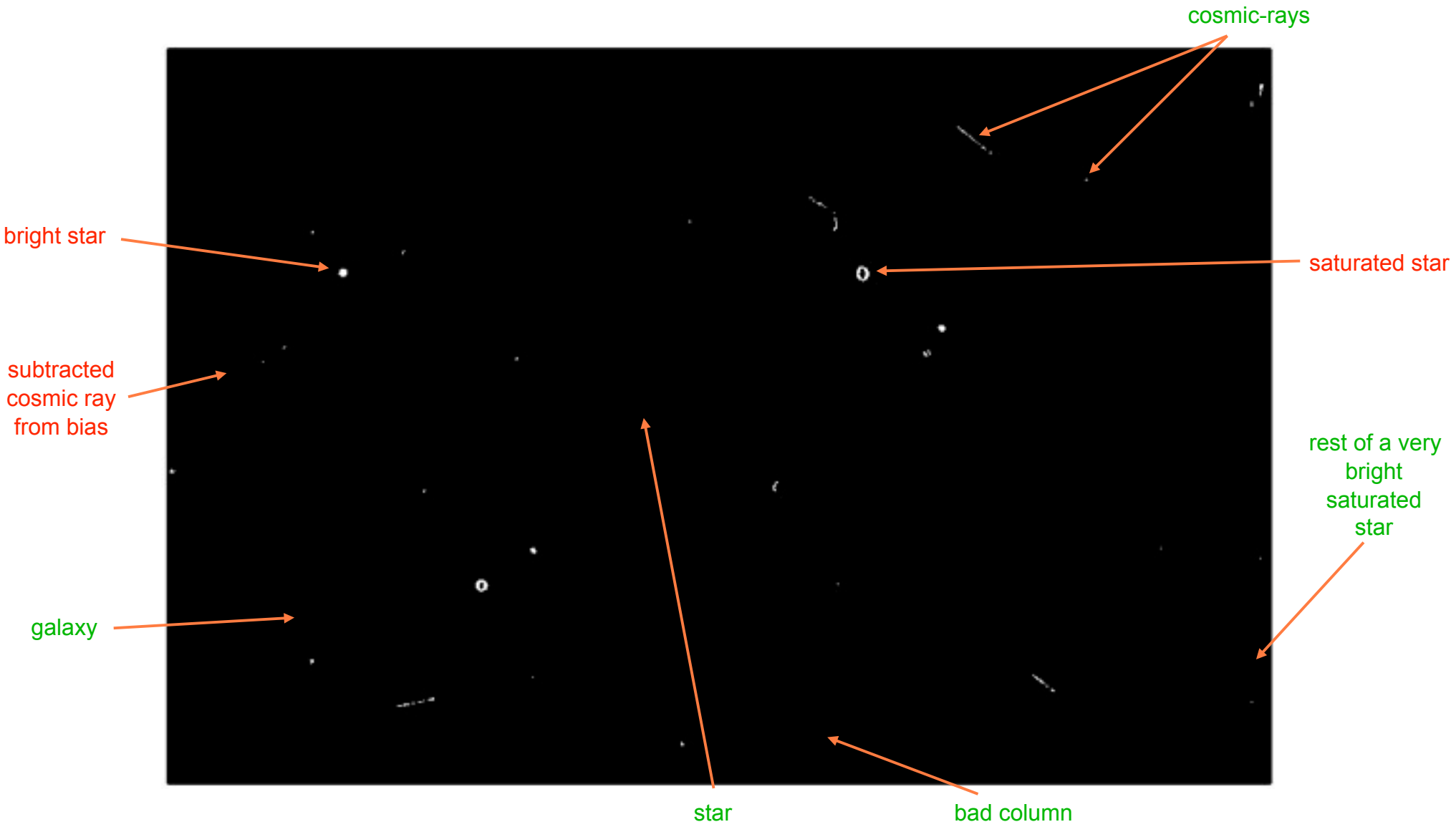
Results in our python code with DES GSN data

> Input image (Overscan, Bias, Flat field and CCD cosmetics “corrected”)



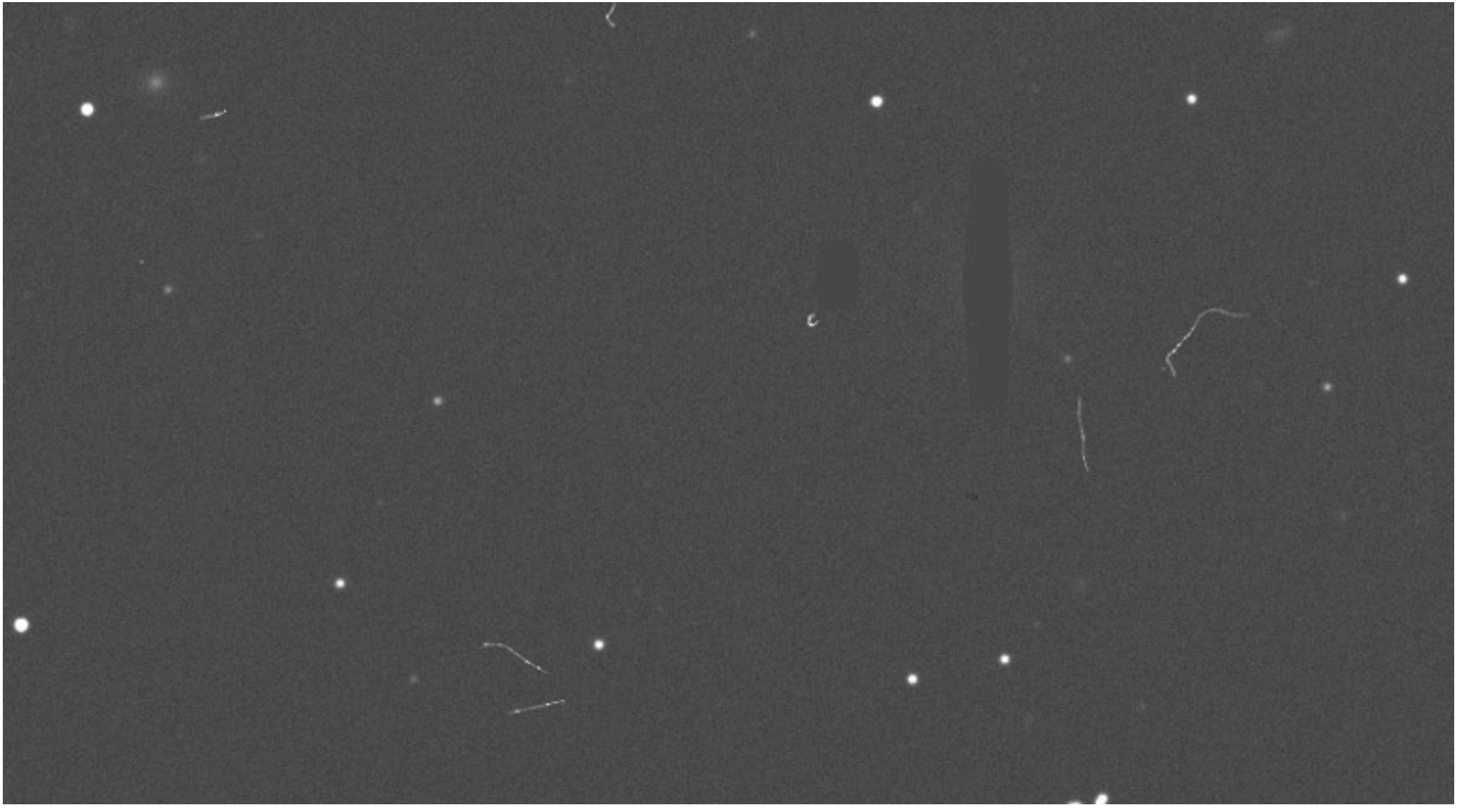
Results in our python code with DES GSN data

> Output Cosmic-Ray Mask



Results in our python code with DES GSN data

> For newer revisions of the code, bright stars are eclipsed at the input image



Further Work

- Optimize parameters for detection (specially f_lim and sigma_lim)
- Avoid masking non CR objects (at least non saturated stars!).
- Reduce CPU processing time. Implement a faster Laplacian and median filtering.
- Perform a better reconstruction method