Scanamorphos port to JAVA

MAX-PLANCK-GESELLSCHAFT

Max-Planck-Institut für extraterrestrische Physik

MPE

Michael Wetzstein

Why port Scanamorphos to Java?

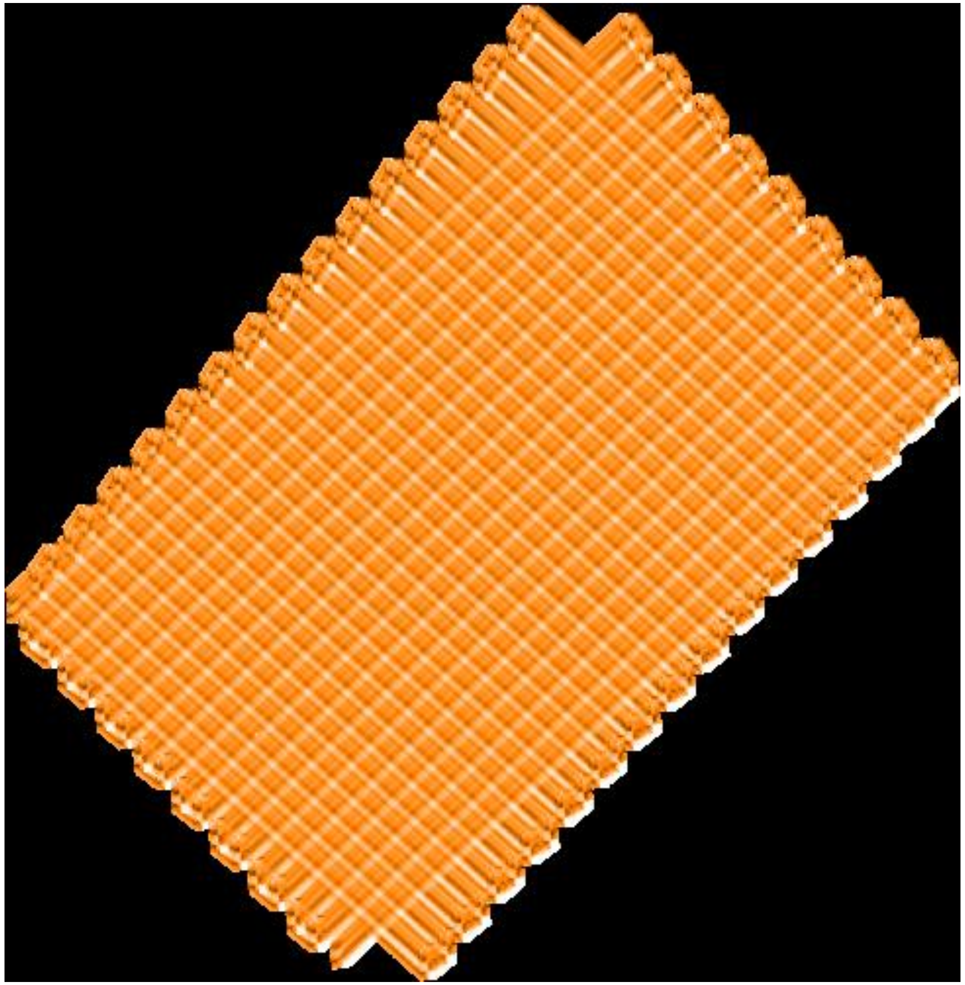
- get a preview of the Scanamorphos results into the Herschel archive
- test our modules by direct comparison with Scanamorphos results

  (well – test Scanamorphos by direct comparison with our results too)
- test the processing idea of Scanamorphos with different code
- understand in detail, how Scanamorphos works

Contents

- remove Turnarounds
- baseline subtraction
- destriping
- remove average drifts (of all detector pixels)
- remove individual drifts (of every individual detector pixel)
- map

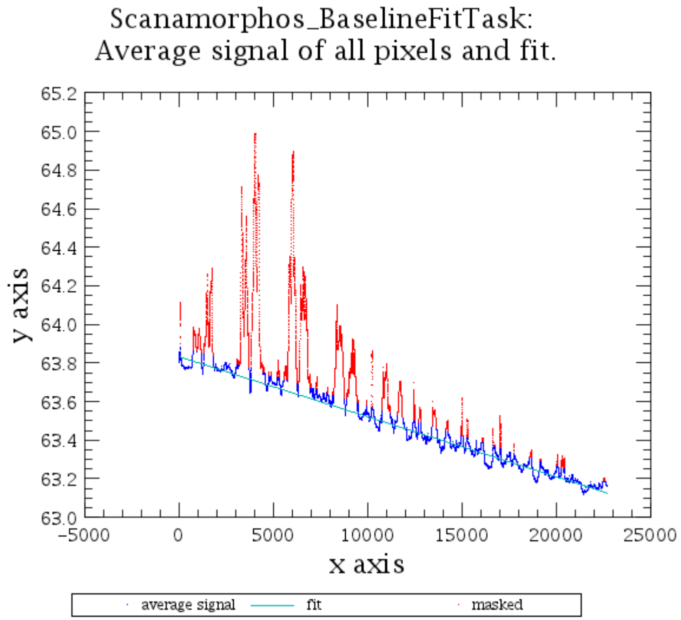
Level 1 of obsid 1342204362 + -63



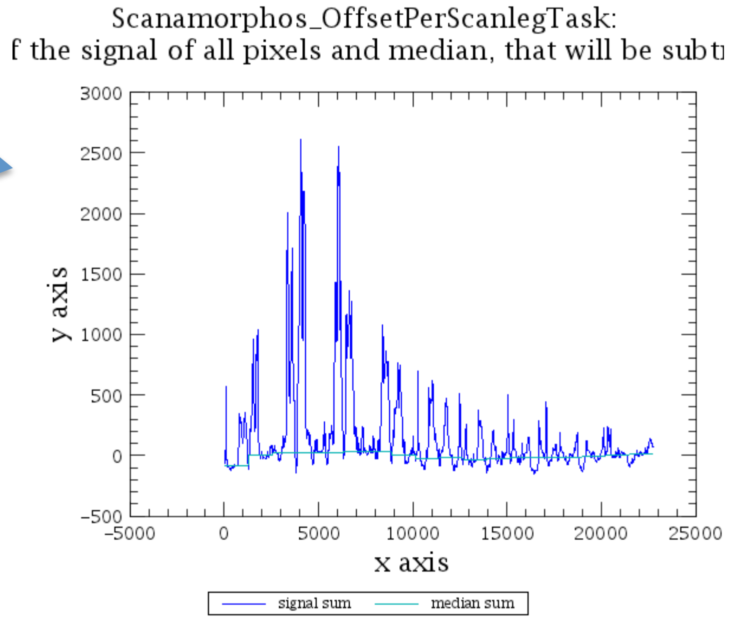
Baseline removal

Repetition by Repetition:

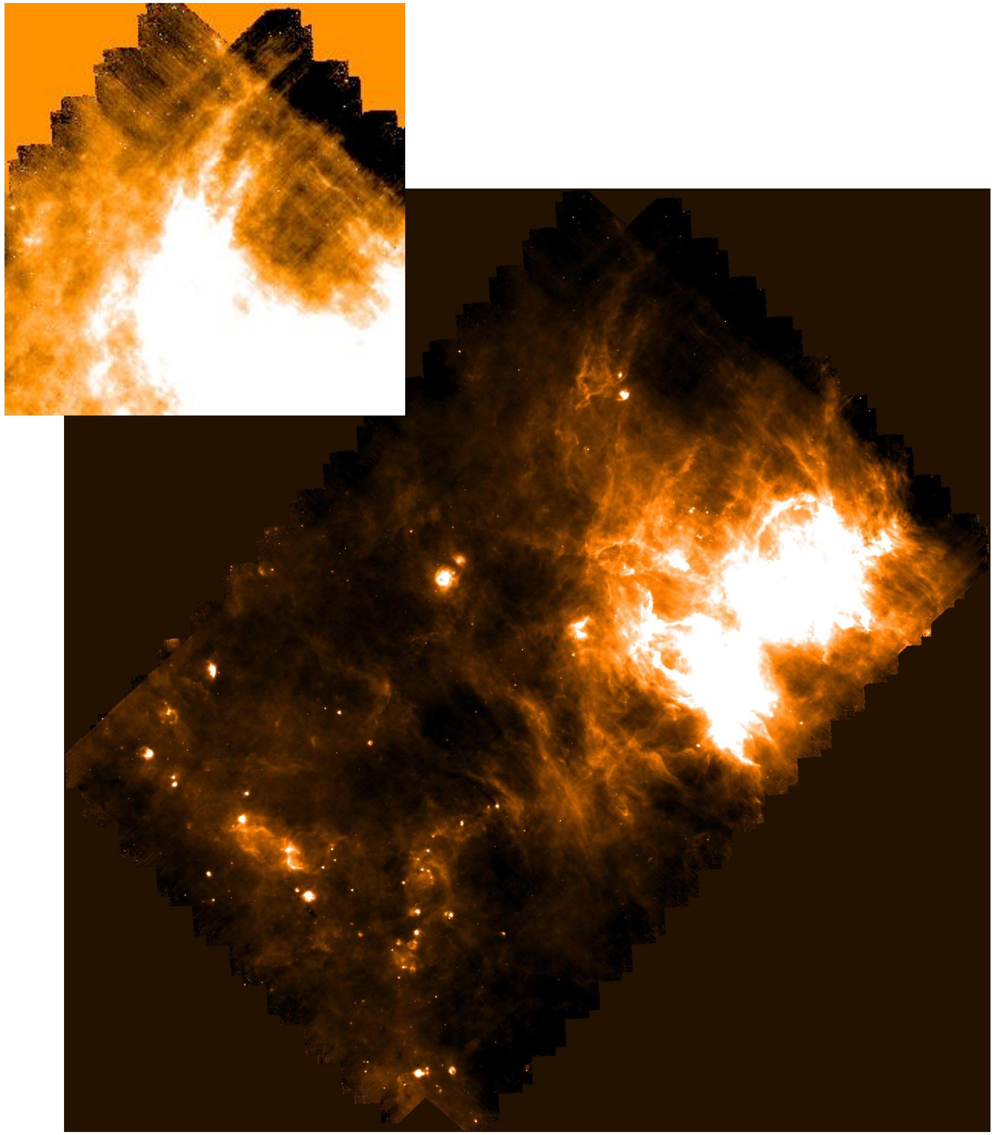
1. baseline fit of average signal, complete timeline. 3-sigma masking

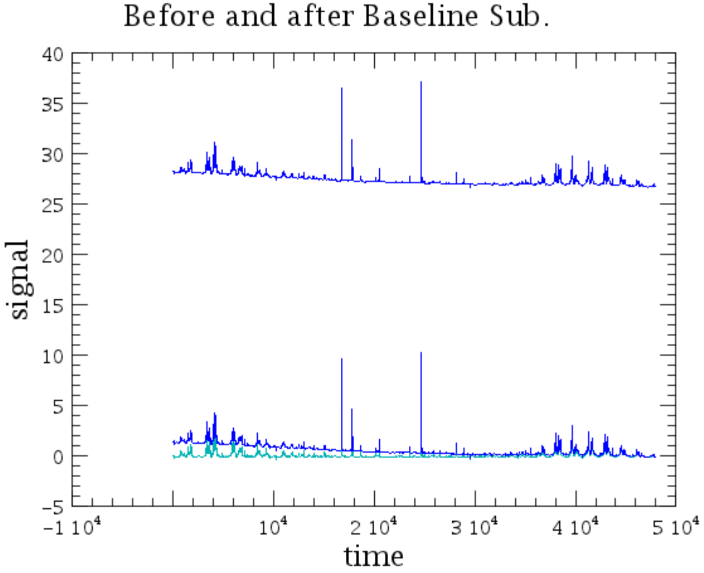


2. subtract the median of all unmasked values per scanleg
3. repeat 1 and 2
4. repeat 1
5. galactic option: repeat 2
6. no galactic: fit scanlegs per detector pixel, 3-sigma masking

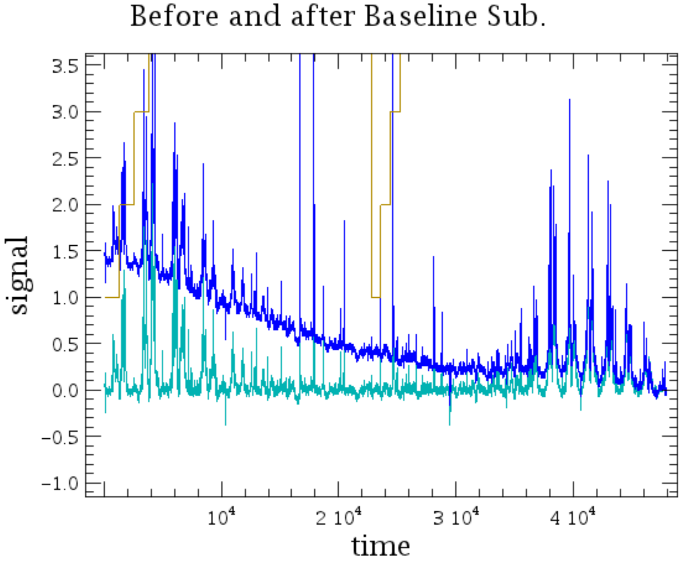


Baseline removal - result



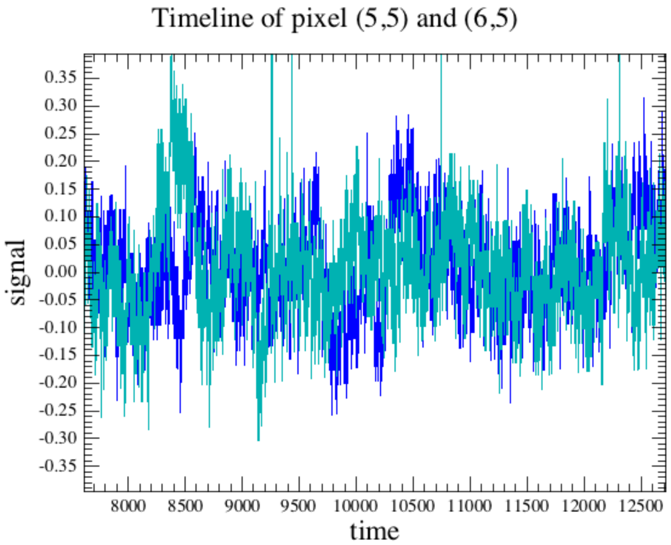


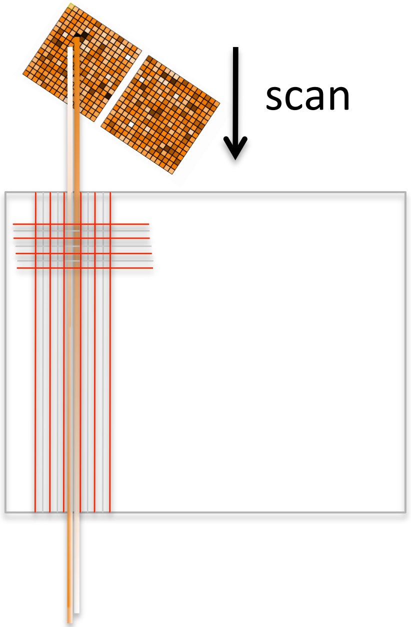
Before and after Baseline Sub.



Before and after Baseline Sub.

Destriping



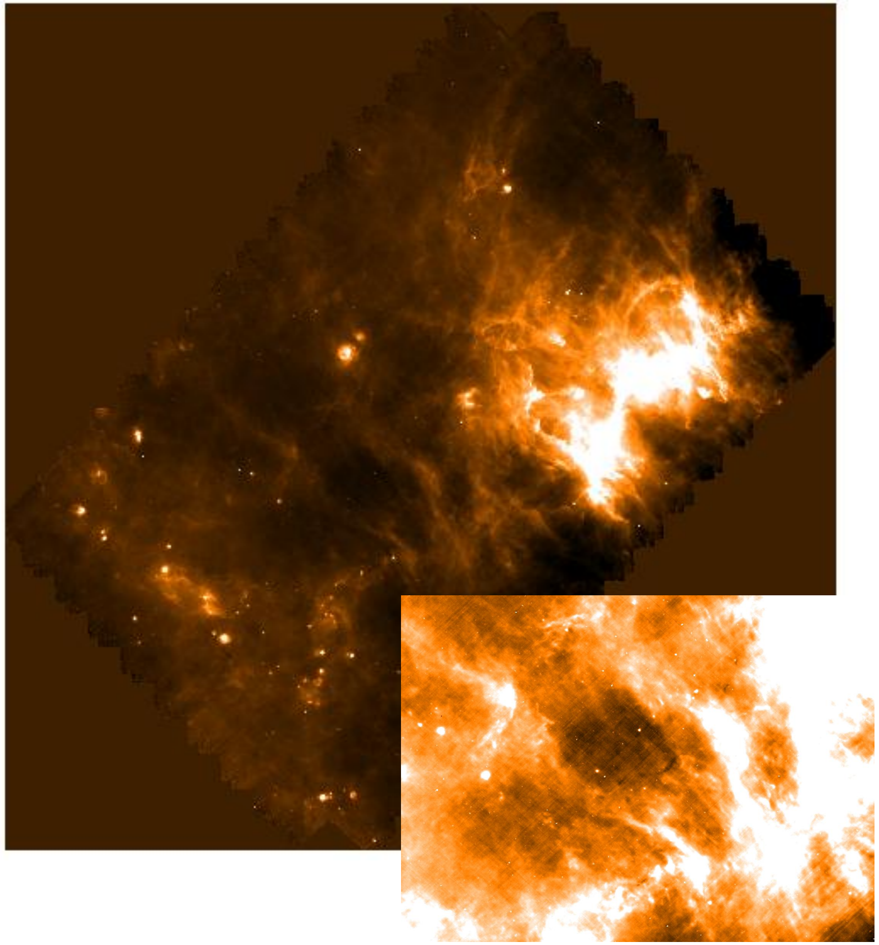


- Takes a Map and backprojects it into a Signal cube

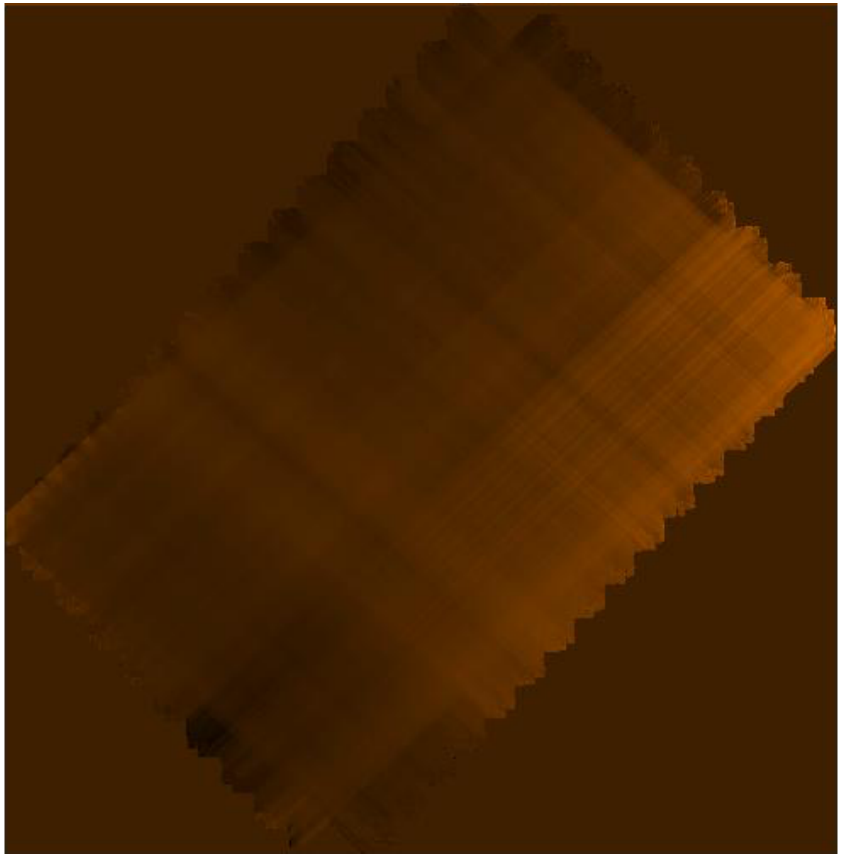
  this should average out drifts
- subract this simulated cube from the real one
- fit the difference
- subtract fit from the signal
- Map
- iterate

Destriping - result

Destriped Map

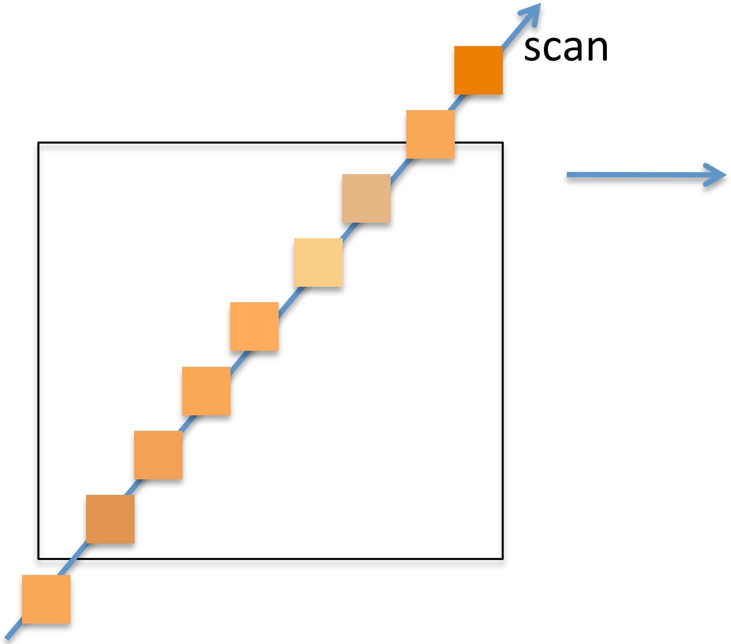


Difference to Baseline-Subtracted Map



Average Drifts

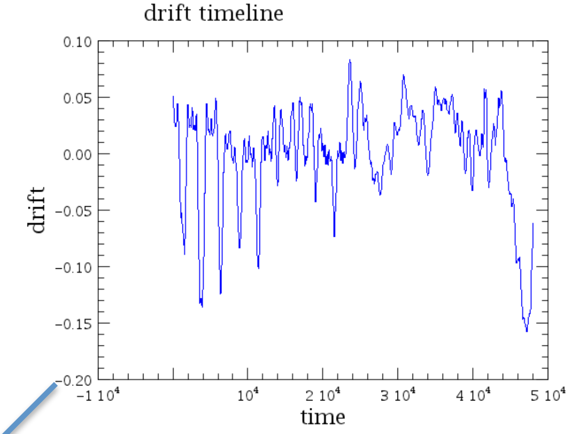
Goal: remove the drift that is common to all detector pixels
Approach:
Collect crossings of Detectors over a Map pixel



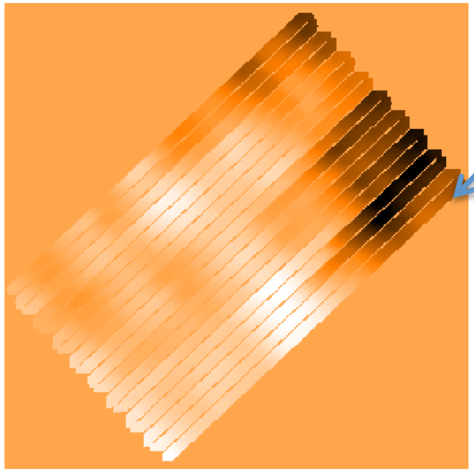
For every Map Pixel:
$\overline{\text{Flux}}_i,\sigma_i$, $\overline{\text{time}}_i$

Build drift matrix: Put Flux_i,σ_i at time_i
Assign $\overline{\text{Flux}}(\text{time}_{k)}$ = 0
Find difference: $\overline{\text{drift}}_{ki} = \overline{\text{Flux}}_k - \overline{\text{Flux}}_i$
Assign $\overline{\text{Flux}}_i$
Iterate over the matrix

Average Drifts 2

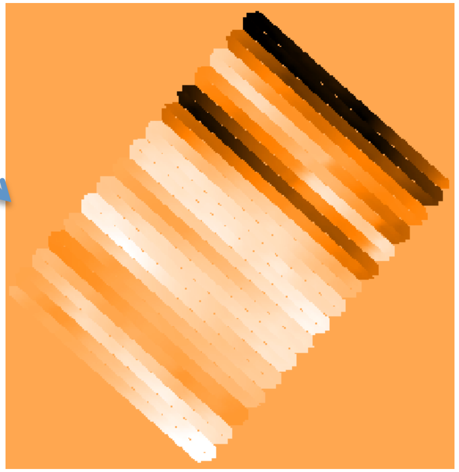

drift timeline

scan



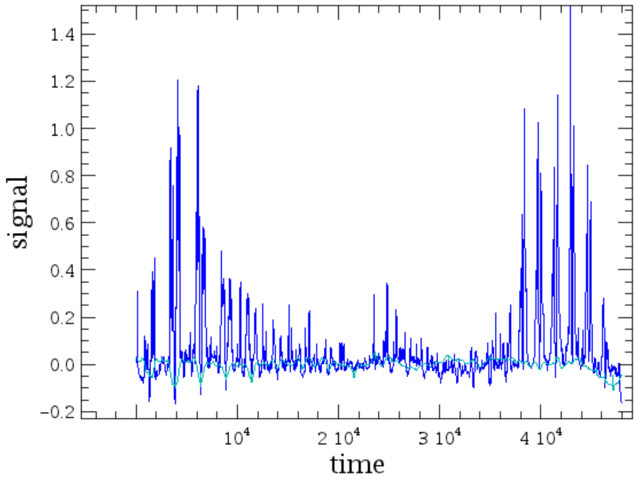
σ-map

cross-scan



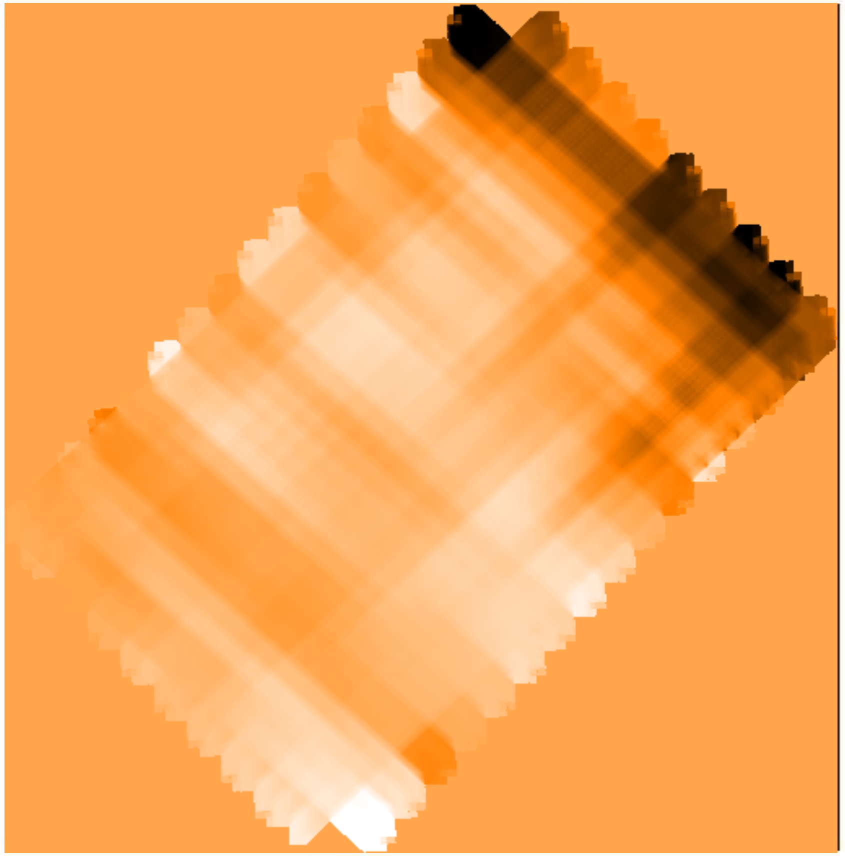
σ-map

Drift removal iteration 1

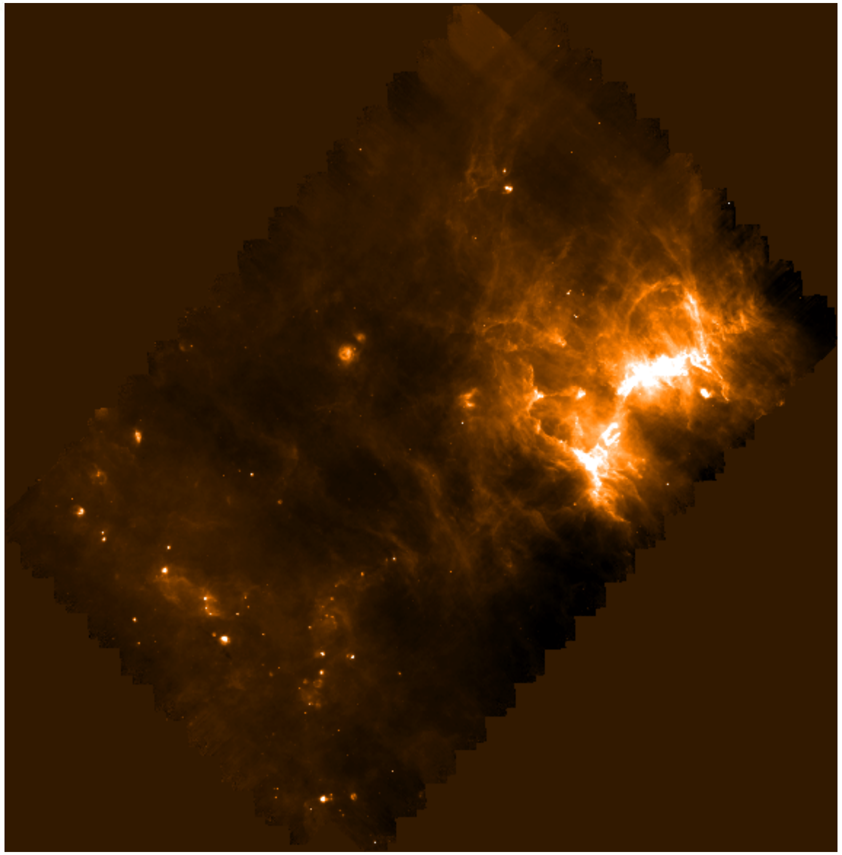


Average - result

Destriped Map

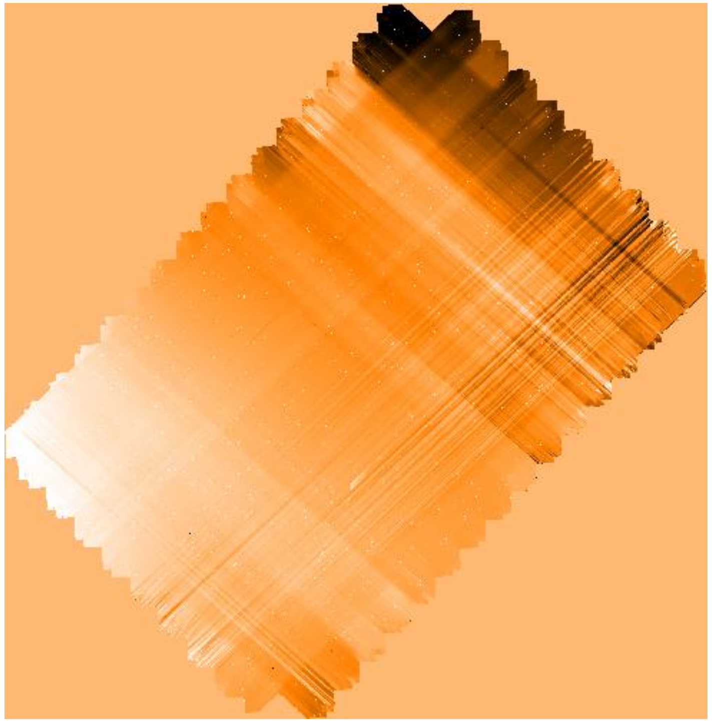


Map after average Drift removal

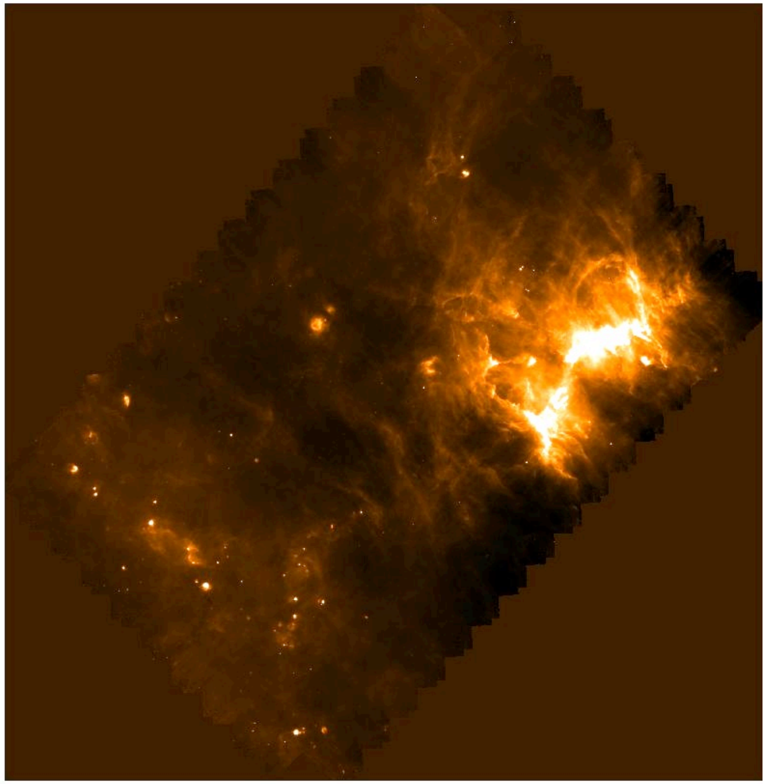


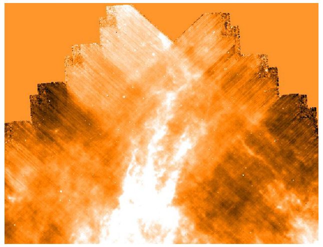
Individual Drifts

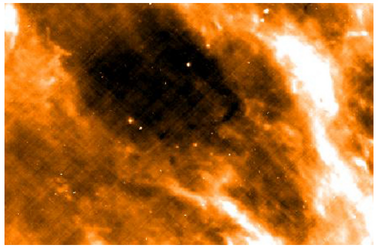
- Calculate the crossings again (like for the average drift)
- Calculate the average flux of all crossings for one Mappixel
- take the difference as drift and assign it to the timeline, this time for every detector pixel
- subtract the drift cube from the signal cube
- iterate



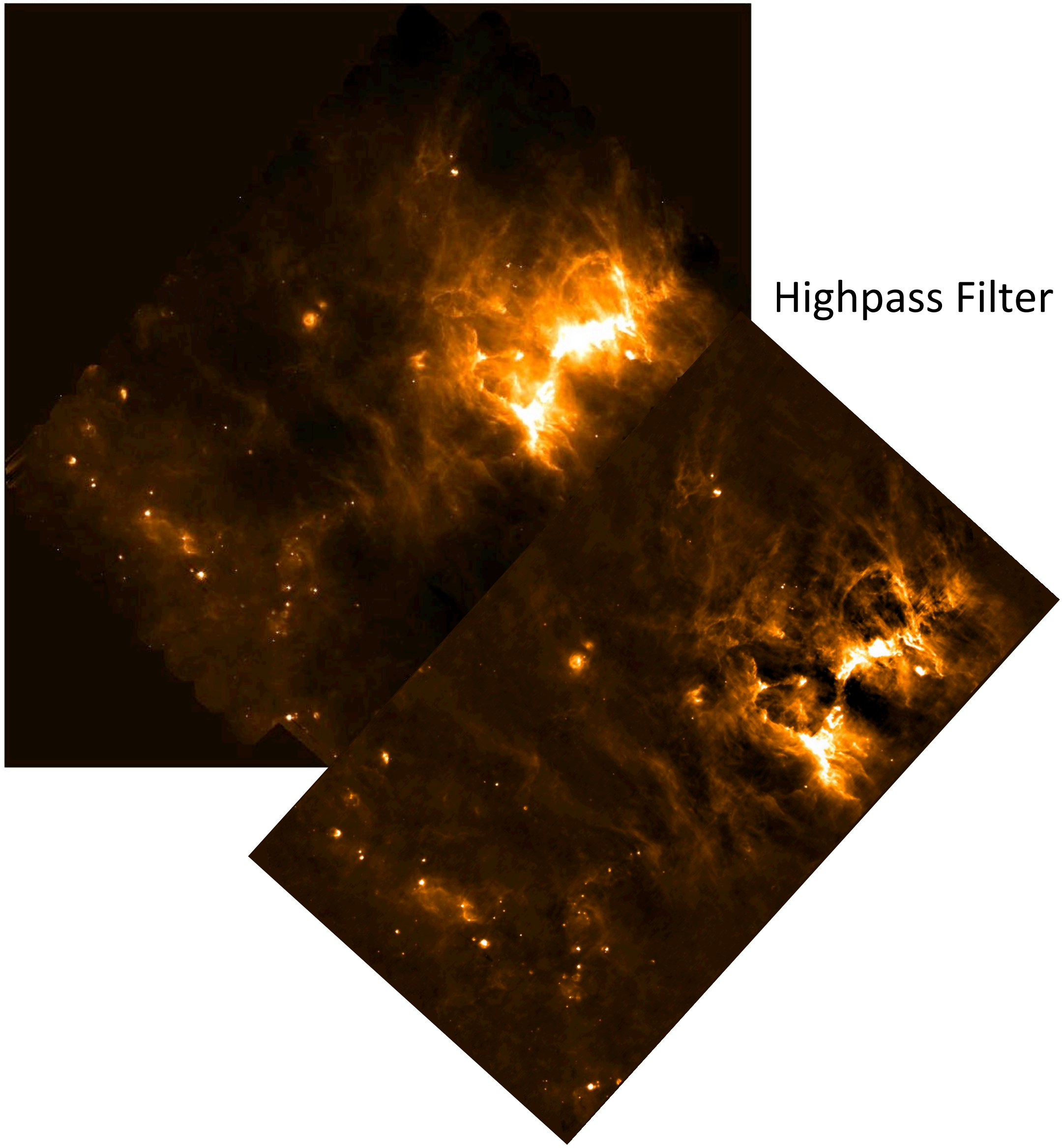
Compare







Madmap



Highpass Filter

Next Steps

- Extended Documentation
- Process all scan-cross-scan pairs
- solve remaining problems (Average drift, perhaps destriper)
- get this ready for the pipeline from software version 9 on

... find a good name