Scanamorphos and JScanam

Max-Planck-Institut für extraterrestrische Physik

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
- understand in detail, how Scanamorphos works

What did we get?

- A Jython wrapper task that calls a handful of (java) tasks -> editable
- an implementation of Scanamorphos' algorithm in hipe

-> accepts level1 data from the archive, returns a map and a frames product

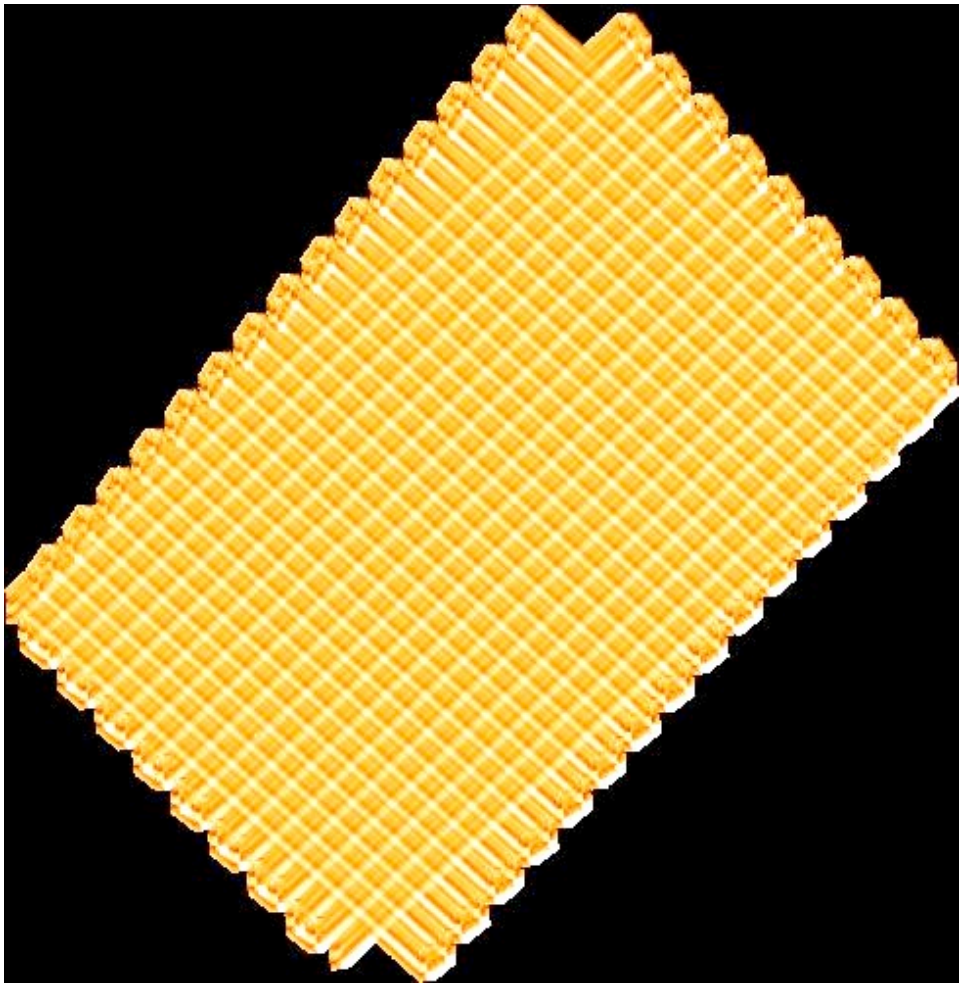
- A document that gives a different view on the algorithm
  http://herschel.esac.esa.int/twiki/bin/view/Pacs/JPScanam#IdD

Contents

Two pieces of software, one algorithm:

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

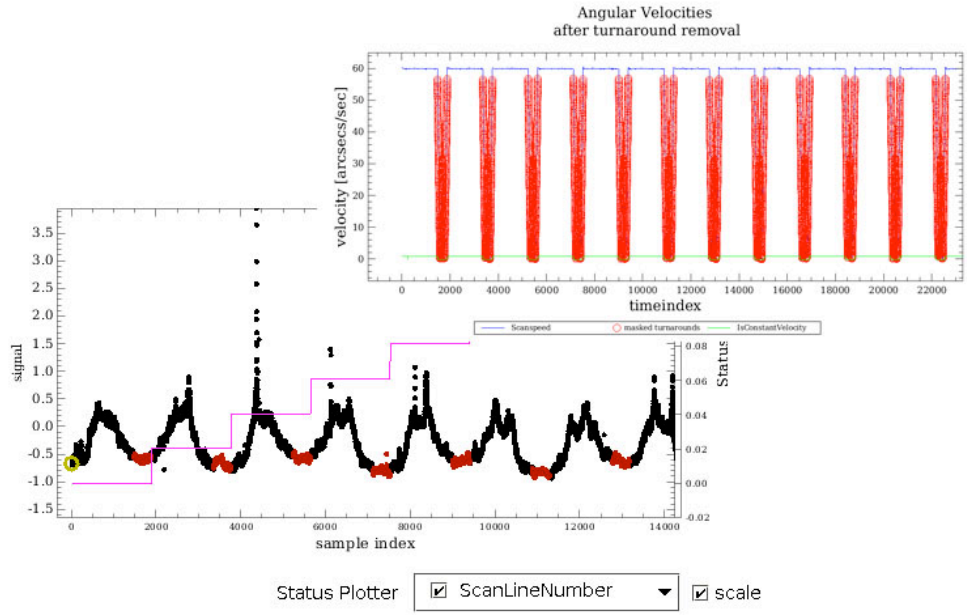
Level 1 of obsid 1342204362 + -63 (M 16)



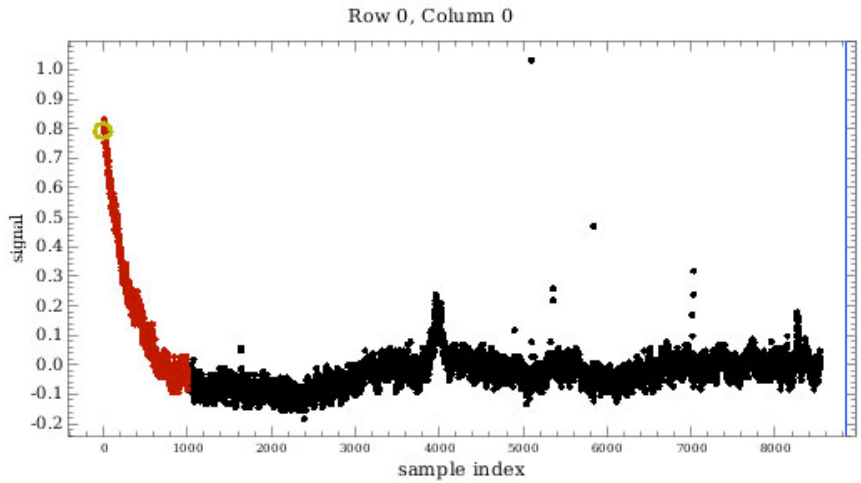
Main idea of the algorithm:
Measure drifts and noise on different scales, fit and take them out.

Turnarounds and Preprocessing

Mask turnarounds:
Filter by scanspeed

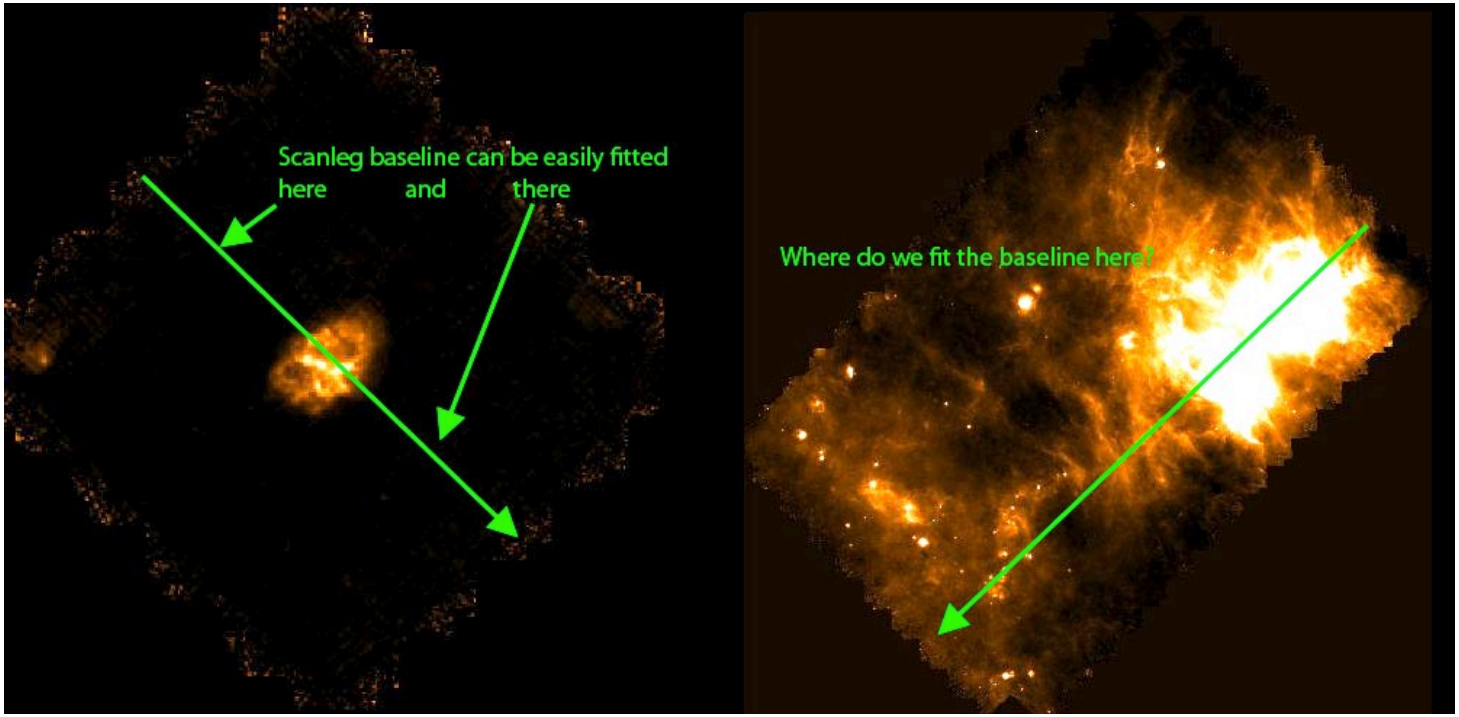


Preprocessing:
Find calibration block relaxation



Baseline removal: galactic or not?

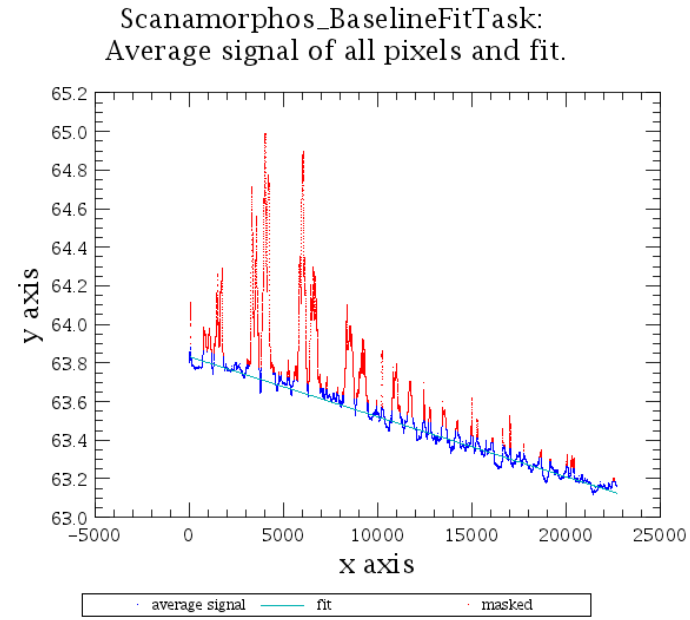
There is a difference between bright compact sources and bright distributed emission:



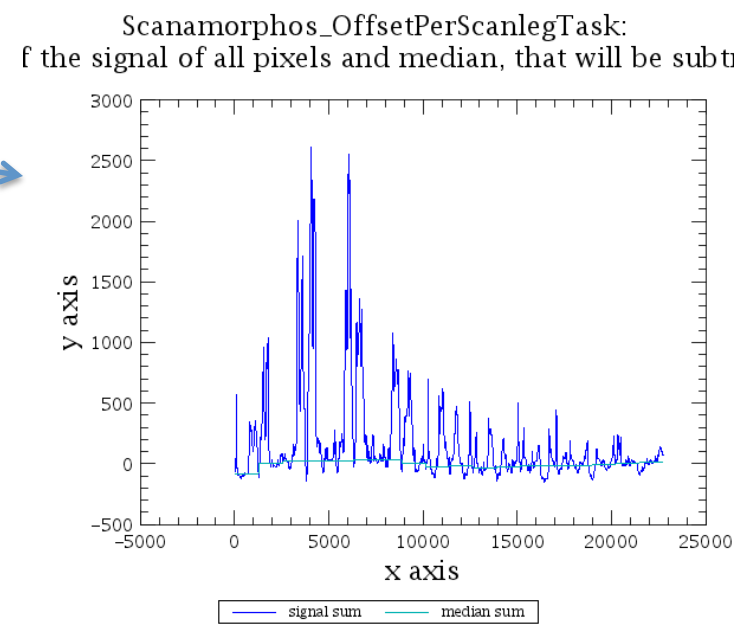
Baseline removal

Repetition by Repetition:
Non-galactic:

1. baseline fit of average signal, complete timeline. 2-sigma masking



2. subtract the median of all unmasked values per scanleg
3. repeat 1 and 2
4. repeat 1
9. fit scanlegs per detector pixel, 2-sigma masking

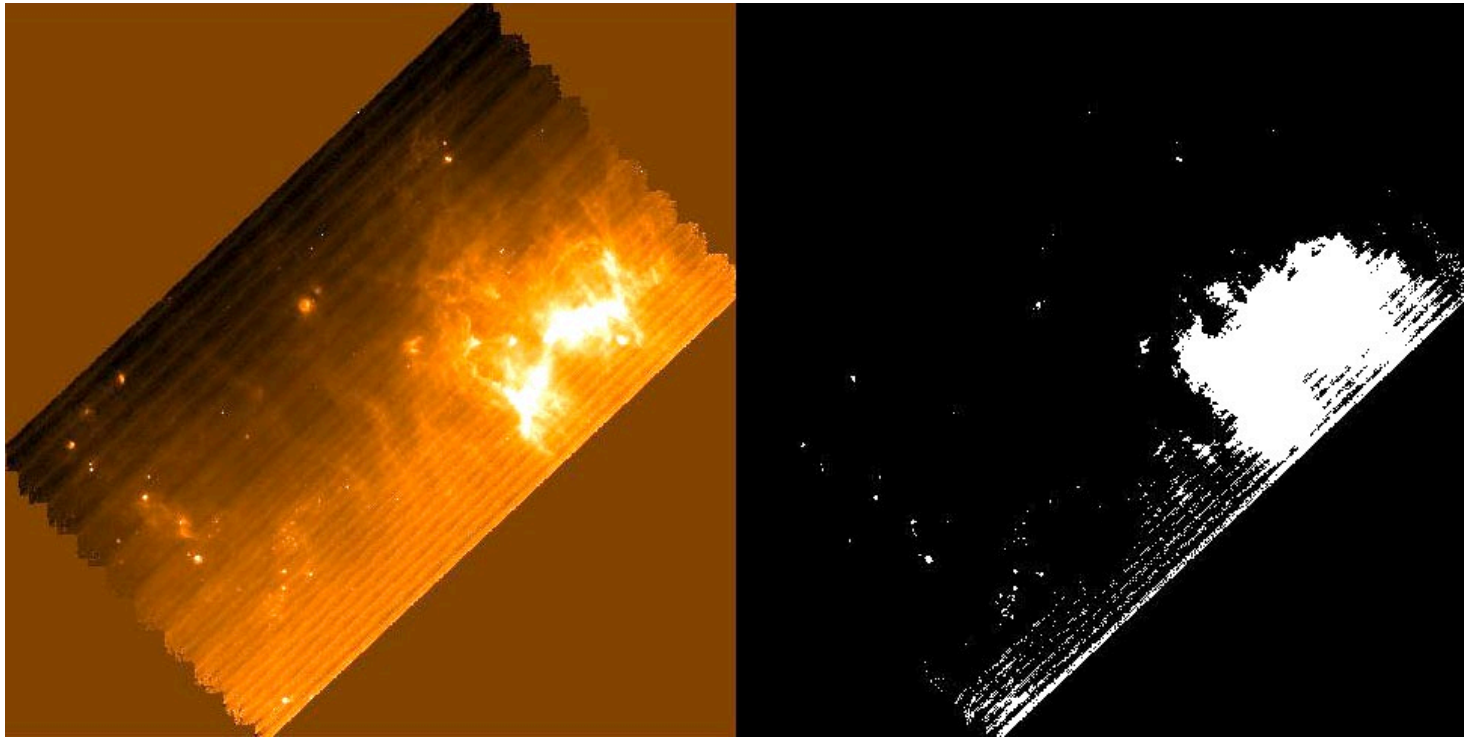


Galactic:

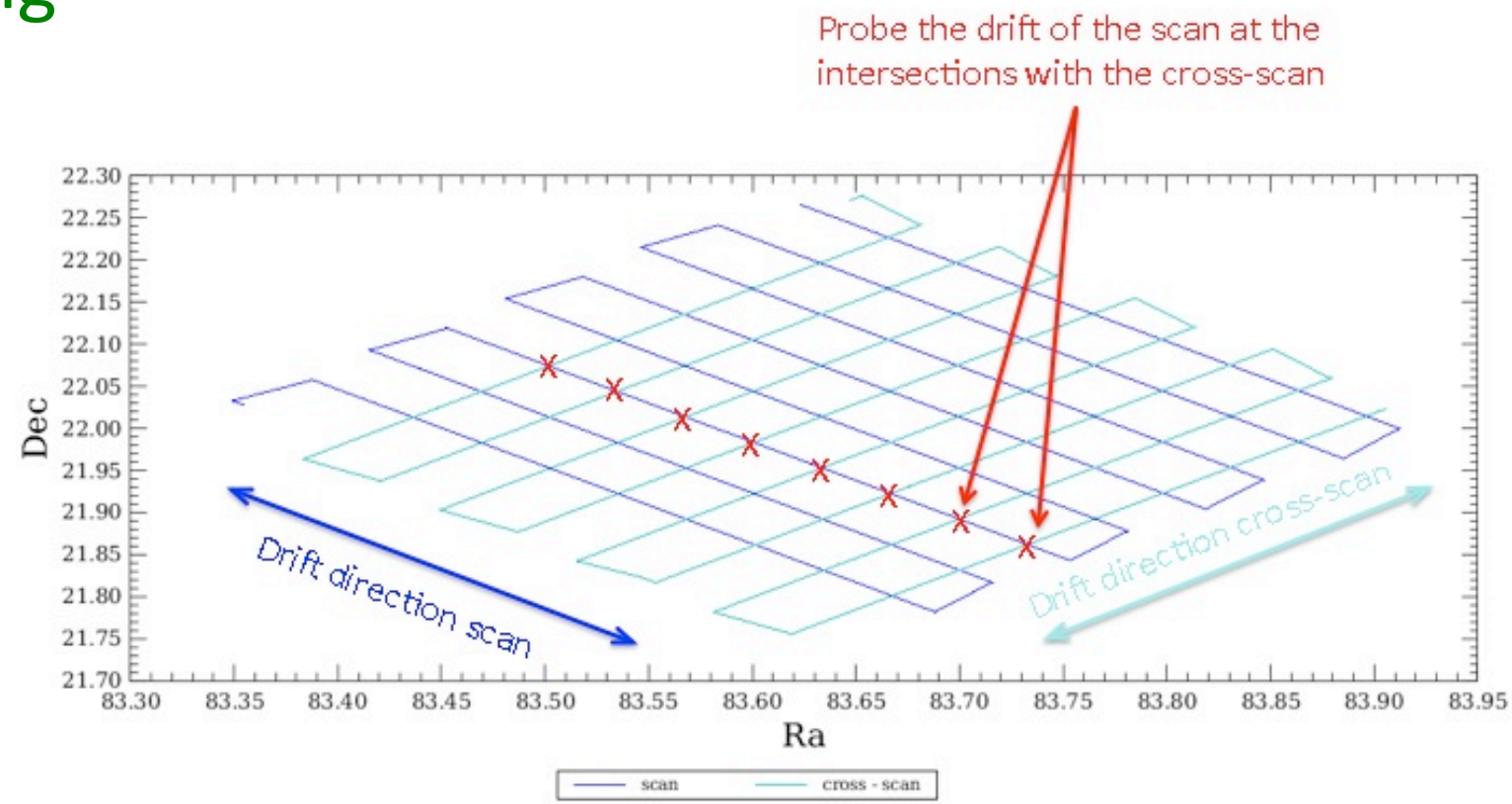
1. only the median recentering is done
2. a threshold sourcemask is calculated
3. median recentering is repeated

Baseline removal - result

Galactic case: map and threshold (source) mask



Destriping

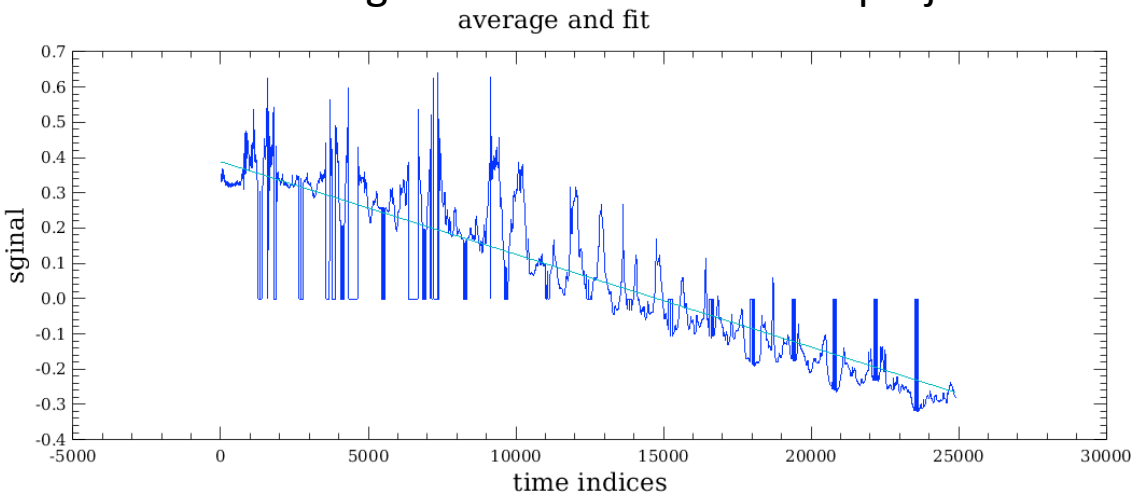


Galactic case: protect real flux gradients!

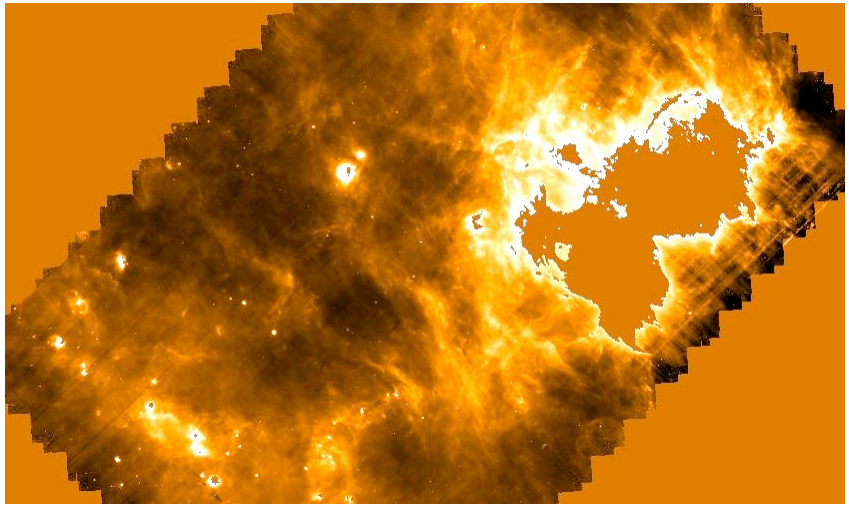
- Fit the backprojected cross-scan map to the scan timelines
- for the average timelines
- for 4 scanleg chunks
- for scanlegs
- refine by back-projecting the full map in later iterations (not only the orthogonal scan)

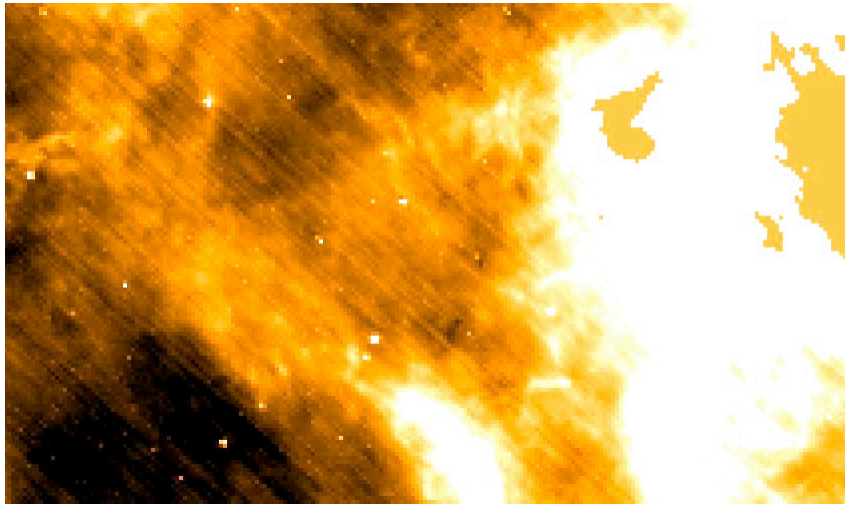
Destriping Results

Fit of the average timeline with the backprojected cross-scan



Detail with short time drifts

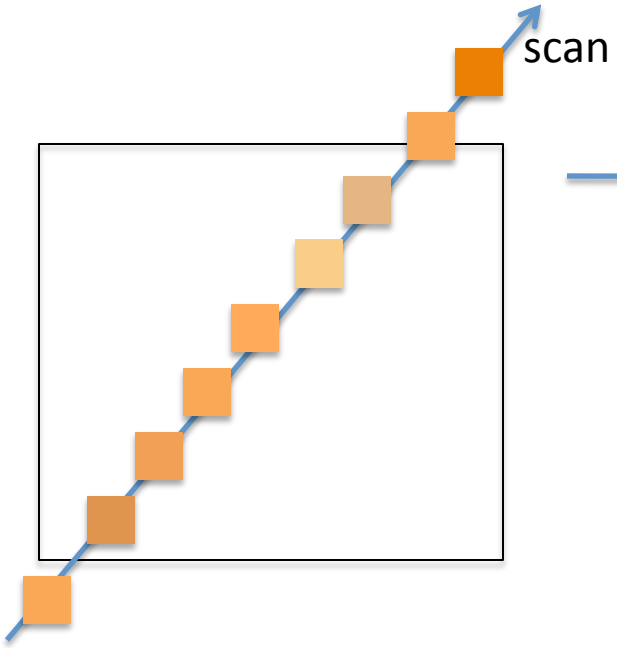




Average Drifts

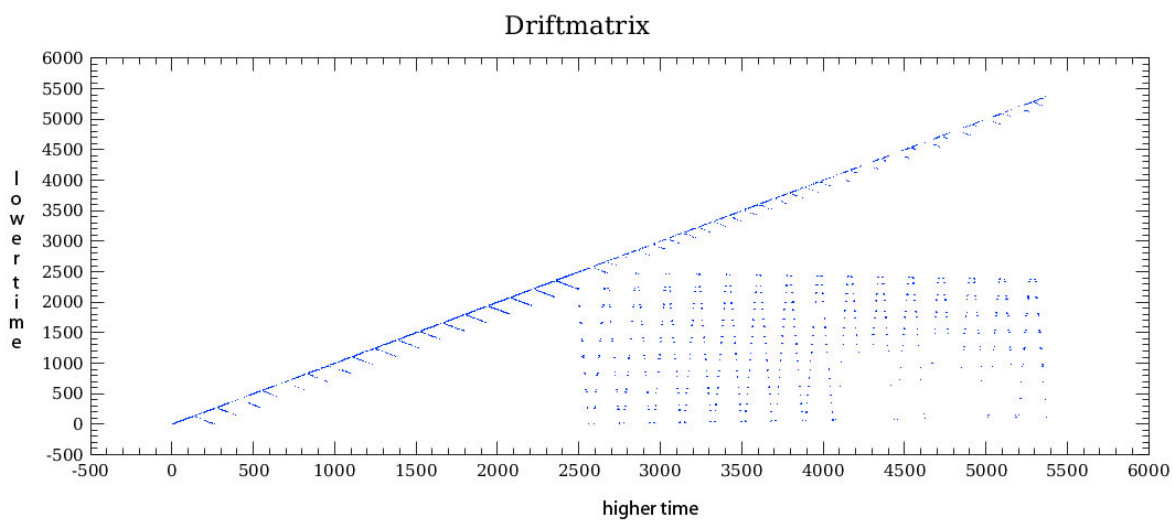
Goal: remove the correlated drift (common to all detector pixels) on a scale smaller than a scanleg

Approach:
Collect crossings of Detectors over a Map pixel

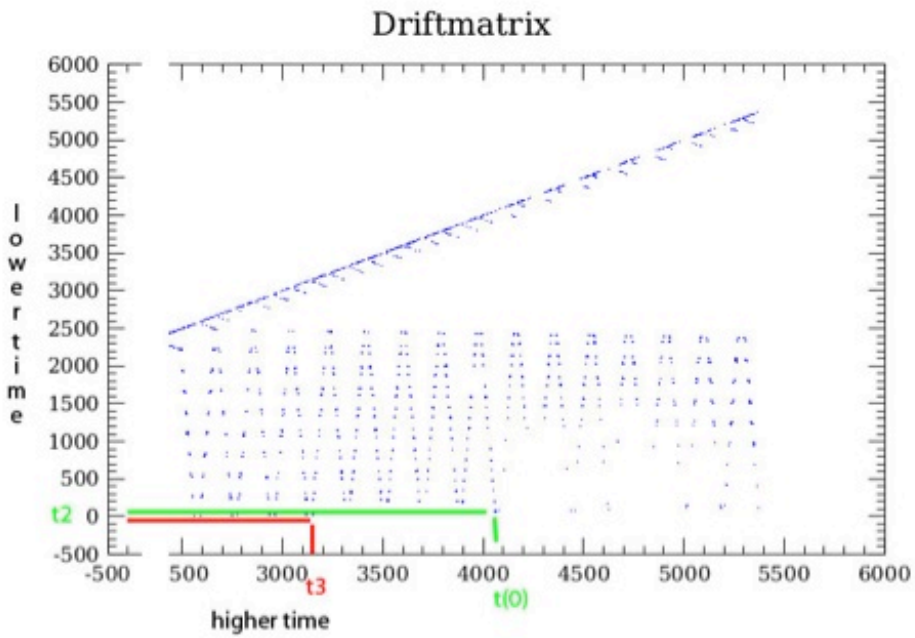


For every Map Pixel:
$\overline{\text{Flux}}_i, \sigma_i, \overline{\text{time}}_i$

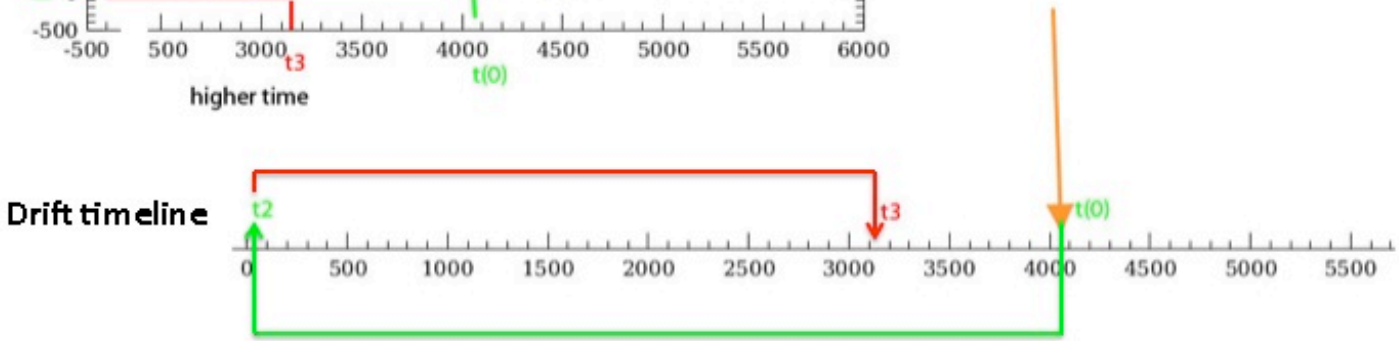
Build drift matrix: Put $\overline{\text{Flux}}_i, \sigma_i$ at $\overline{\text{time}}_i$



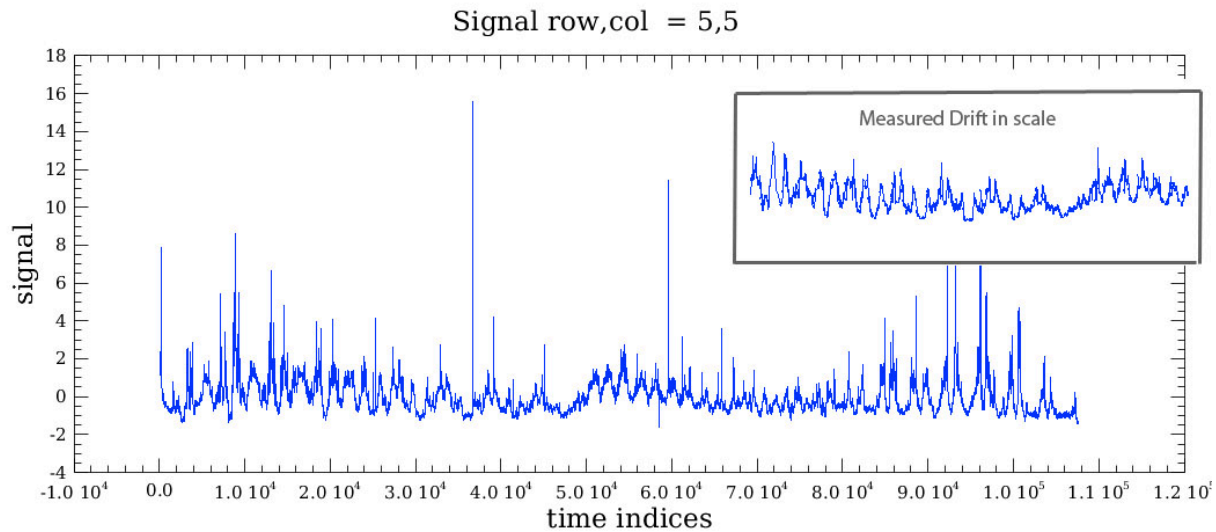
Average Drifts: scan the drift matrix



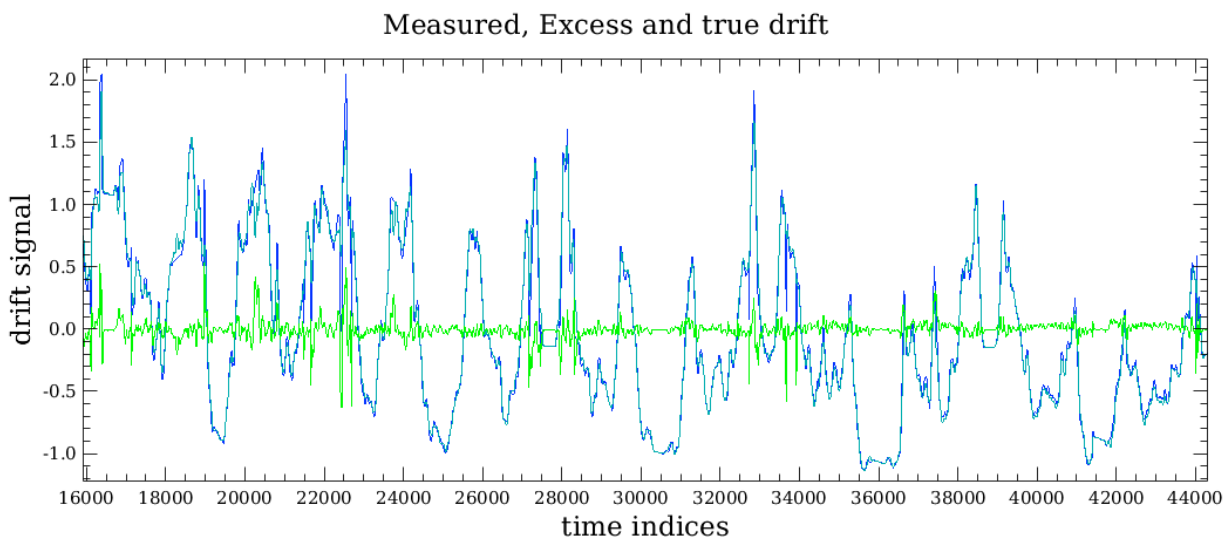
1. Assign drift = 0 (arbitrary)
2. Assign the drift between t(0) and t2
3. find a new high time for low time = t2
4. assign the drift for t3
5. - n. etc.



Average Drifts: find the true drift

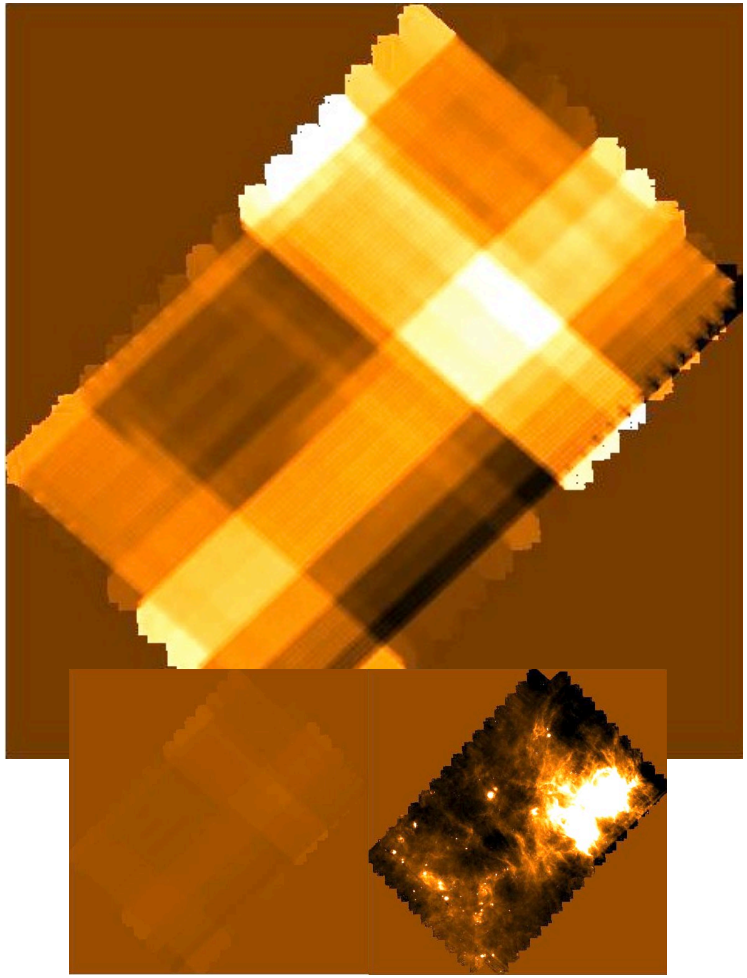


- Scanning the drift matrix inserts an excess drift
- find it by mapping the drift per scan
- weight the maps to get rid of drift, that is special to a repetition
- backproject the result and get the excess drift

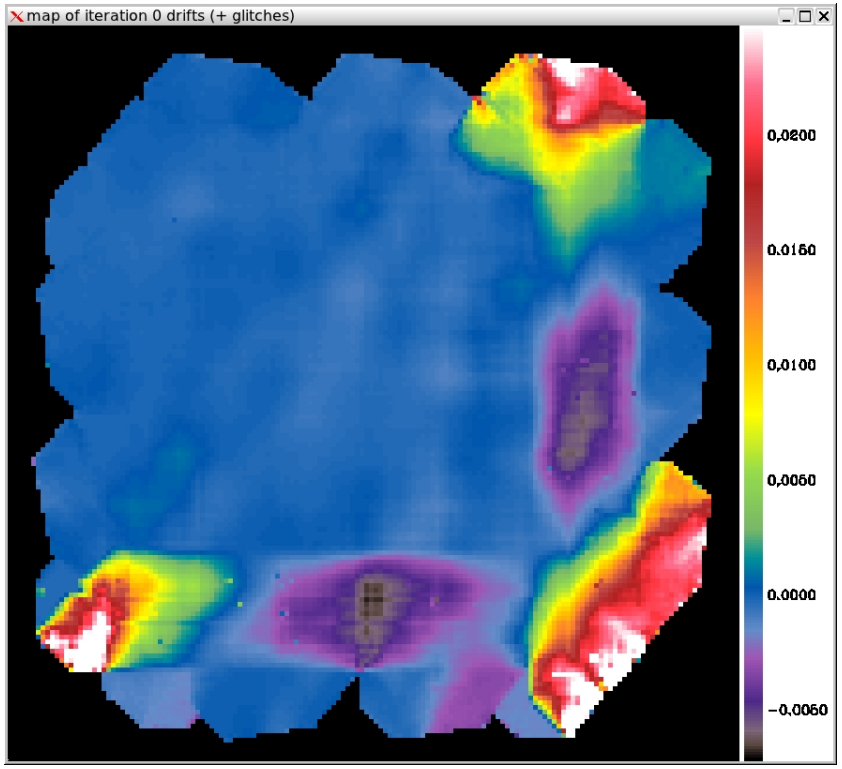


Average - result

JScanam:
Map of the true drift
-> offsets from baselines

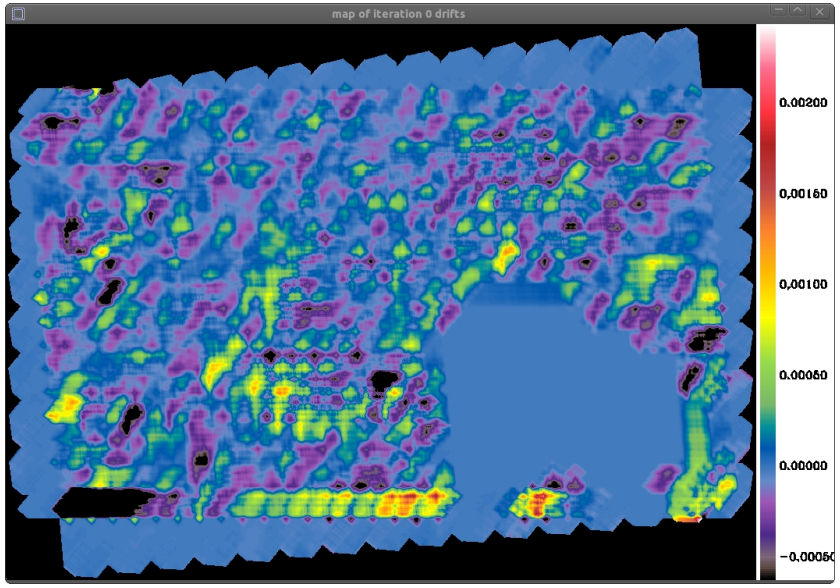


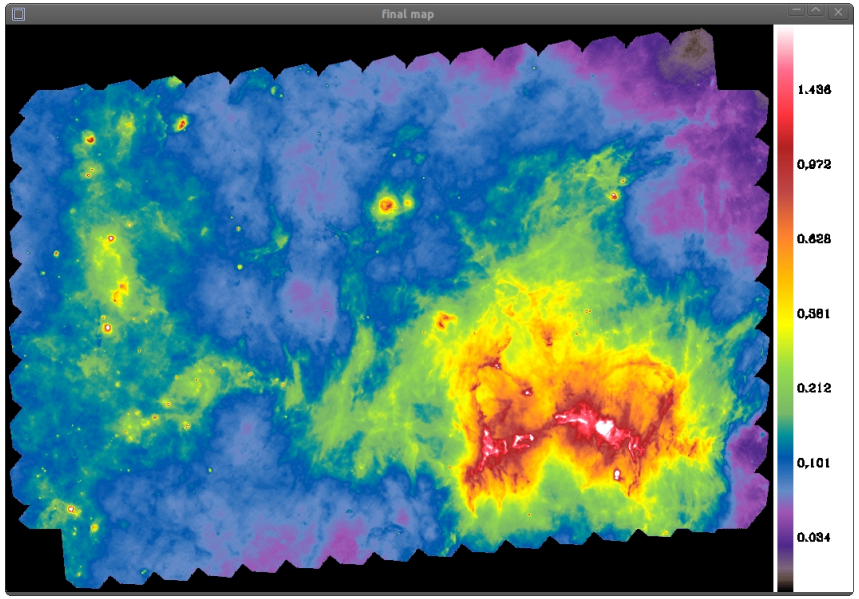
Scanamorphos:
Map of the true drift (NGC 1097)
Visible are the remains of the calibration blocks



Average – result from Scanamorphos

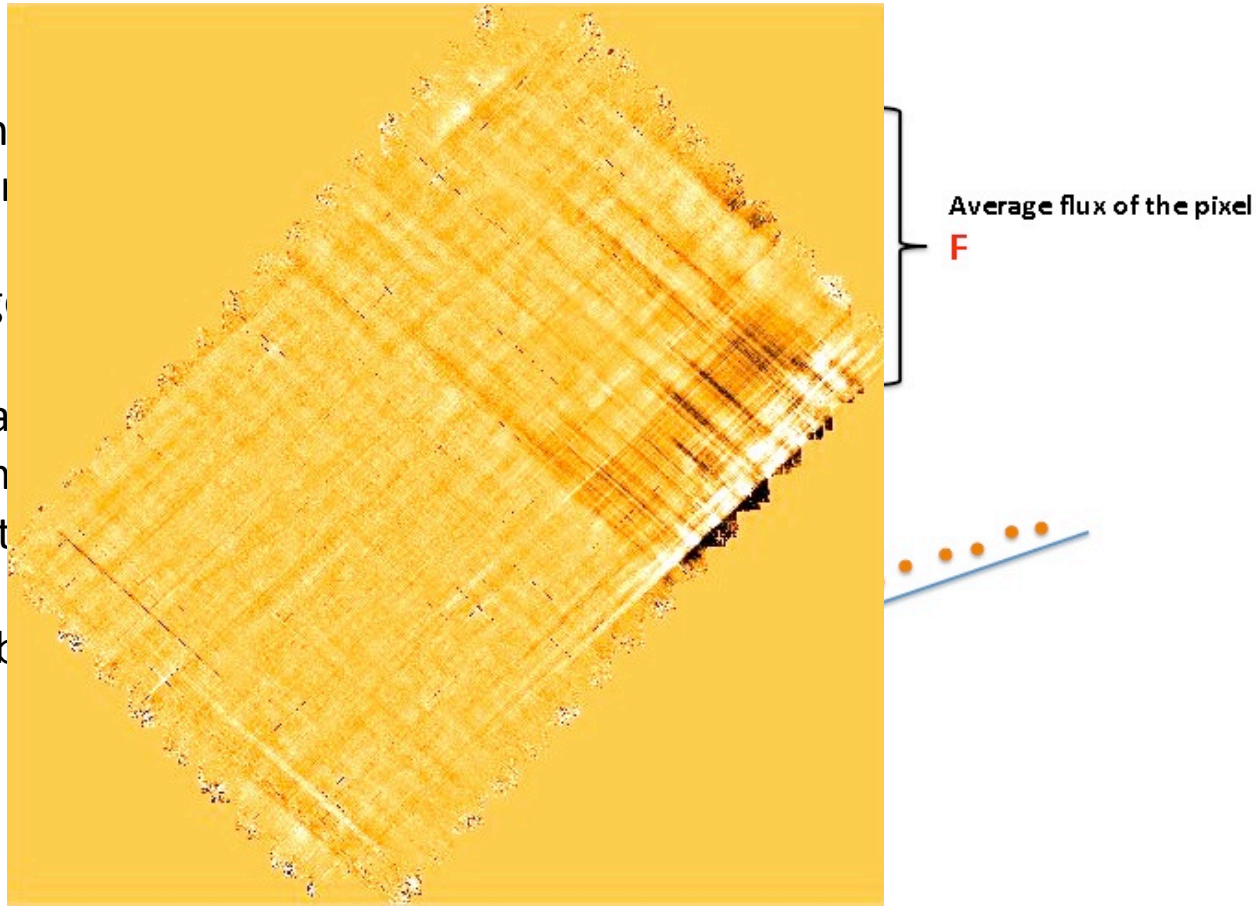
Map of the true drift from Scanamorphos (without baseline artefacts): average drift is also here on a 1% level compared to the signal.



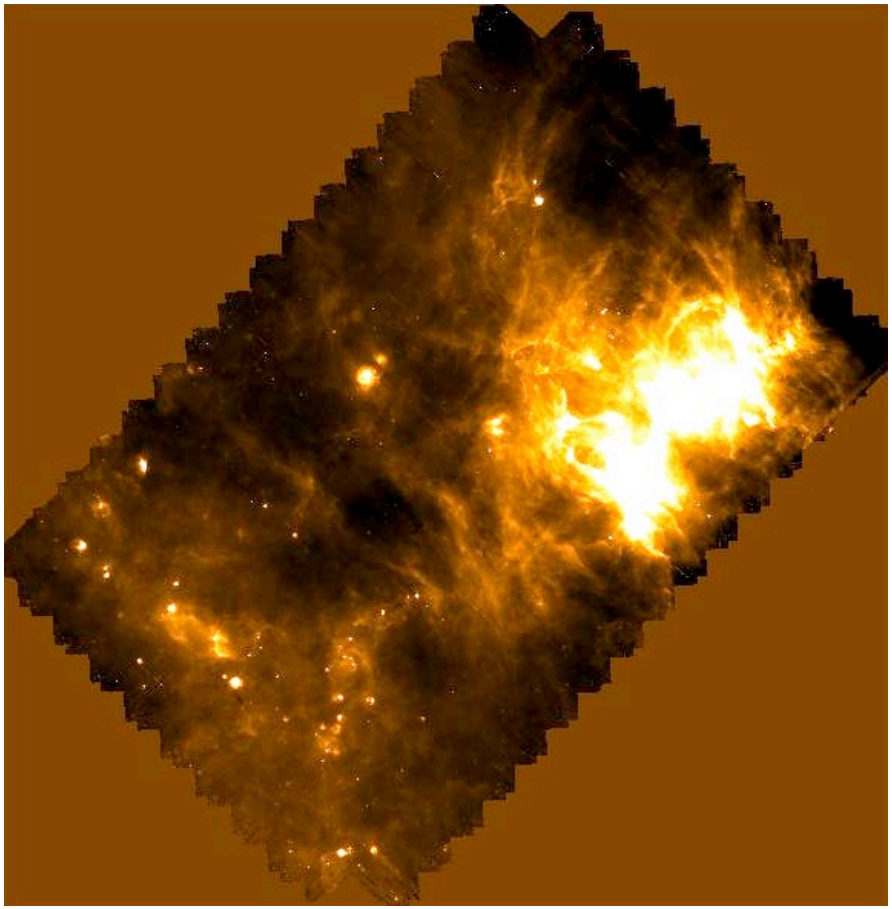


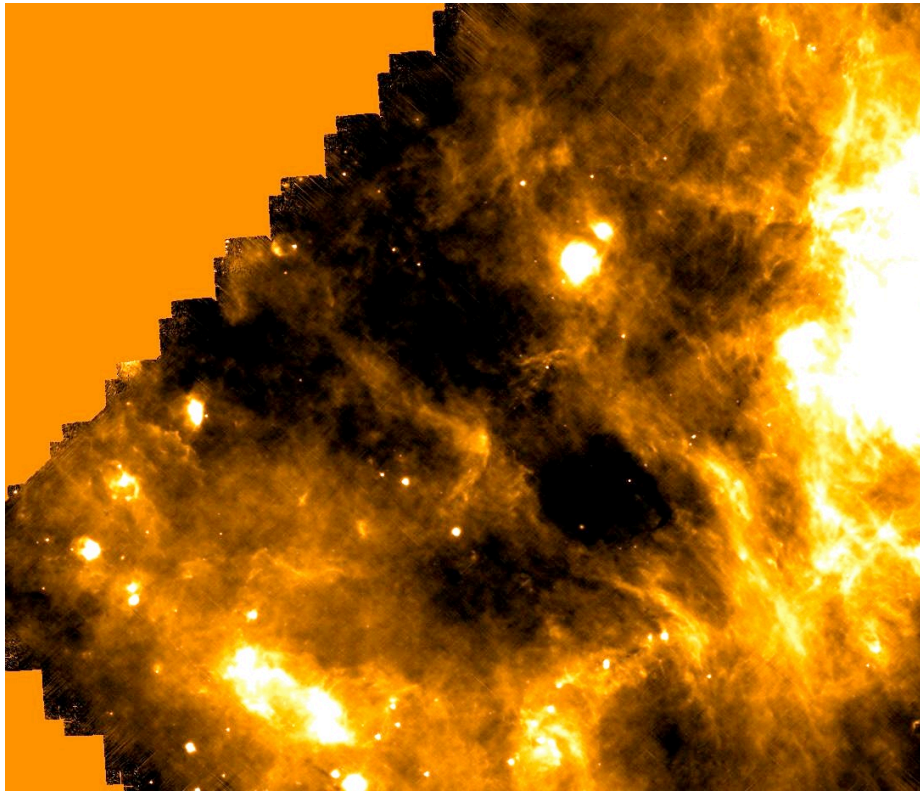
Individual Drifts

- Calculate the crossin
again (like for the aver
drift)
- Calculate the averag
for one Mappixel
- take the difference a
and assign it to the tin
this time for every det
pixel
- subtract the drift cub
from the signal cube
- iterate

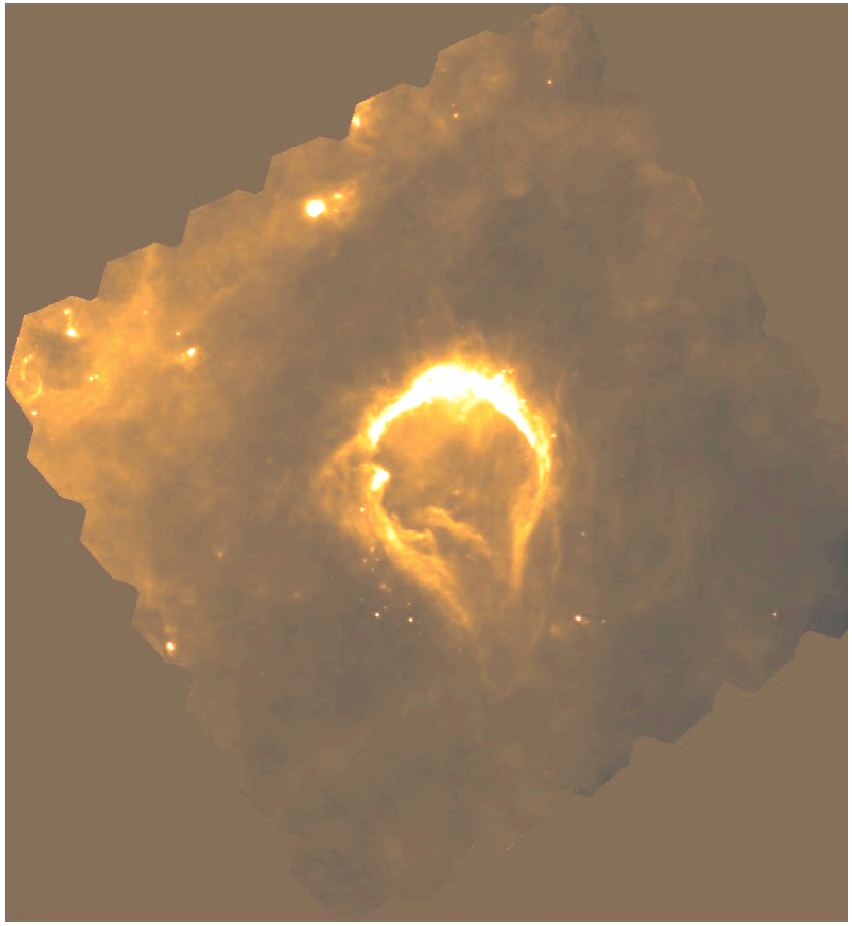


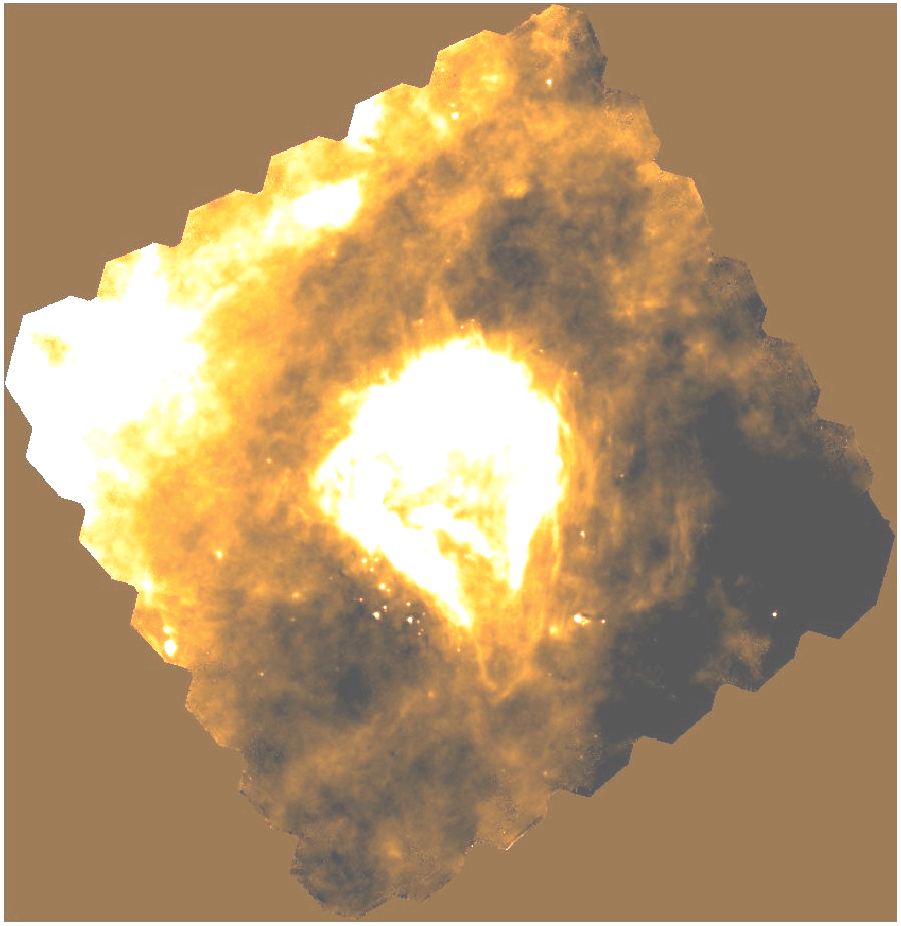
Final Result



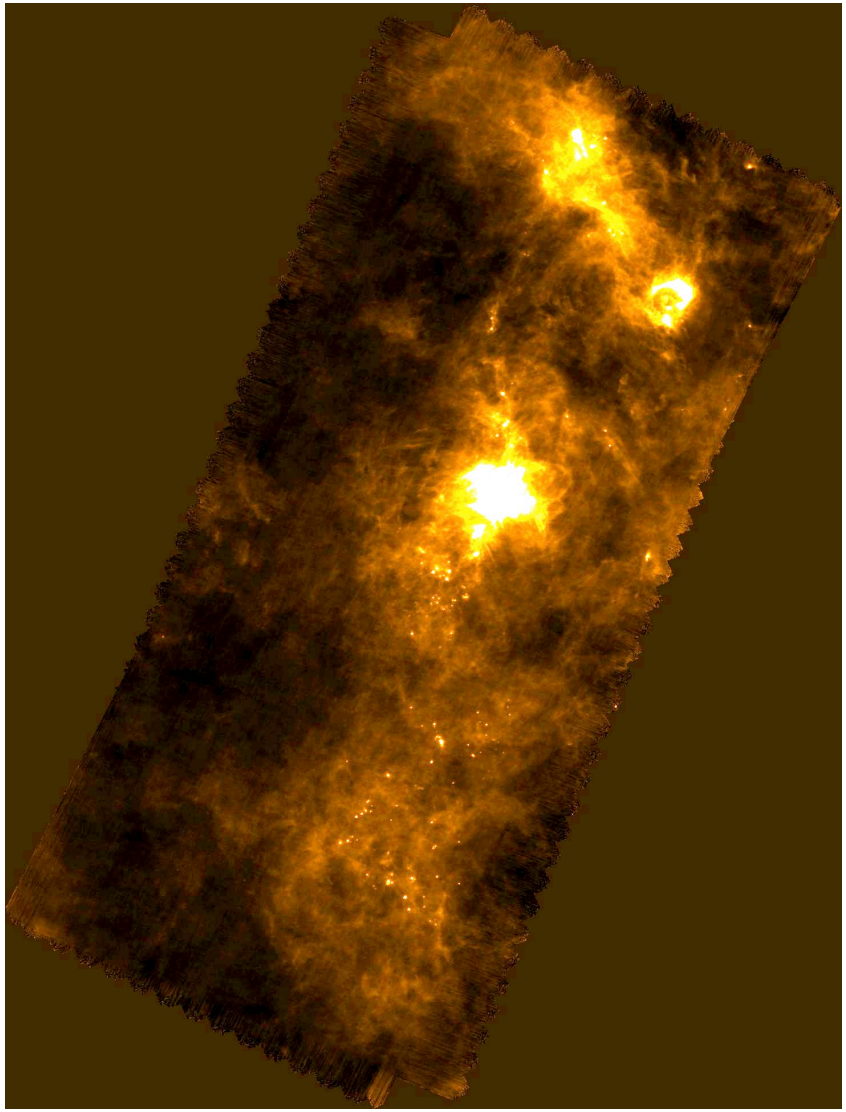


More Examples and Status of JScanam

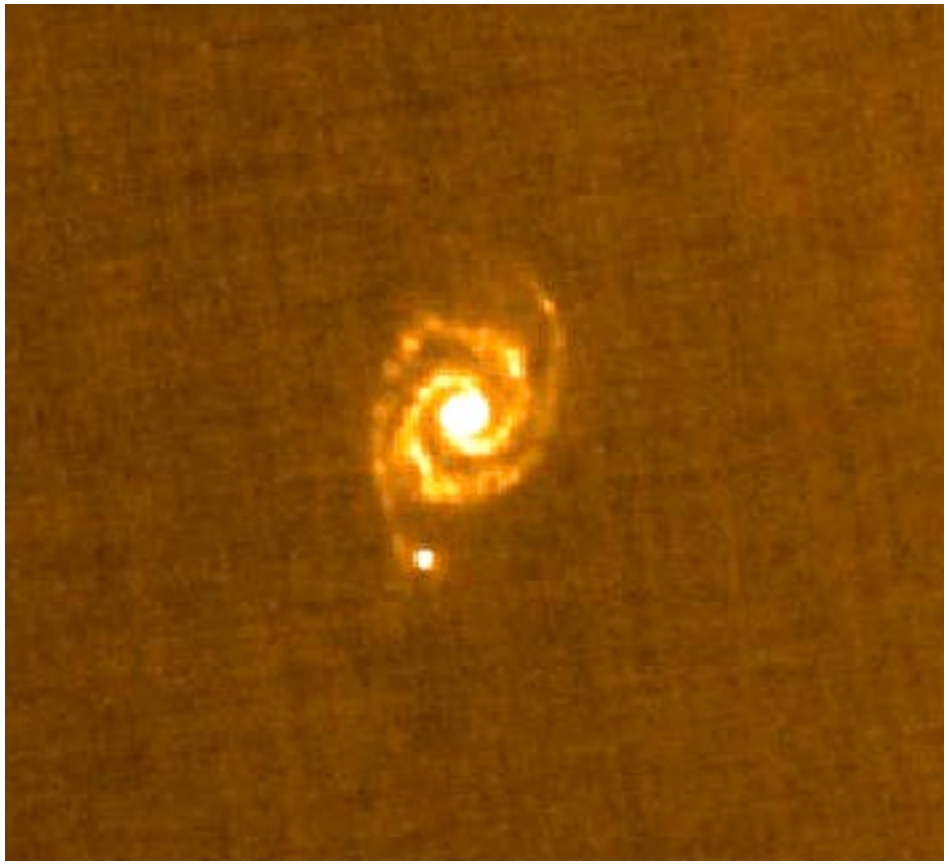
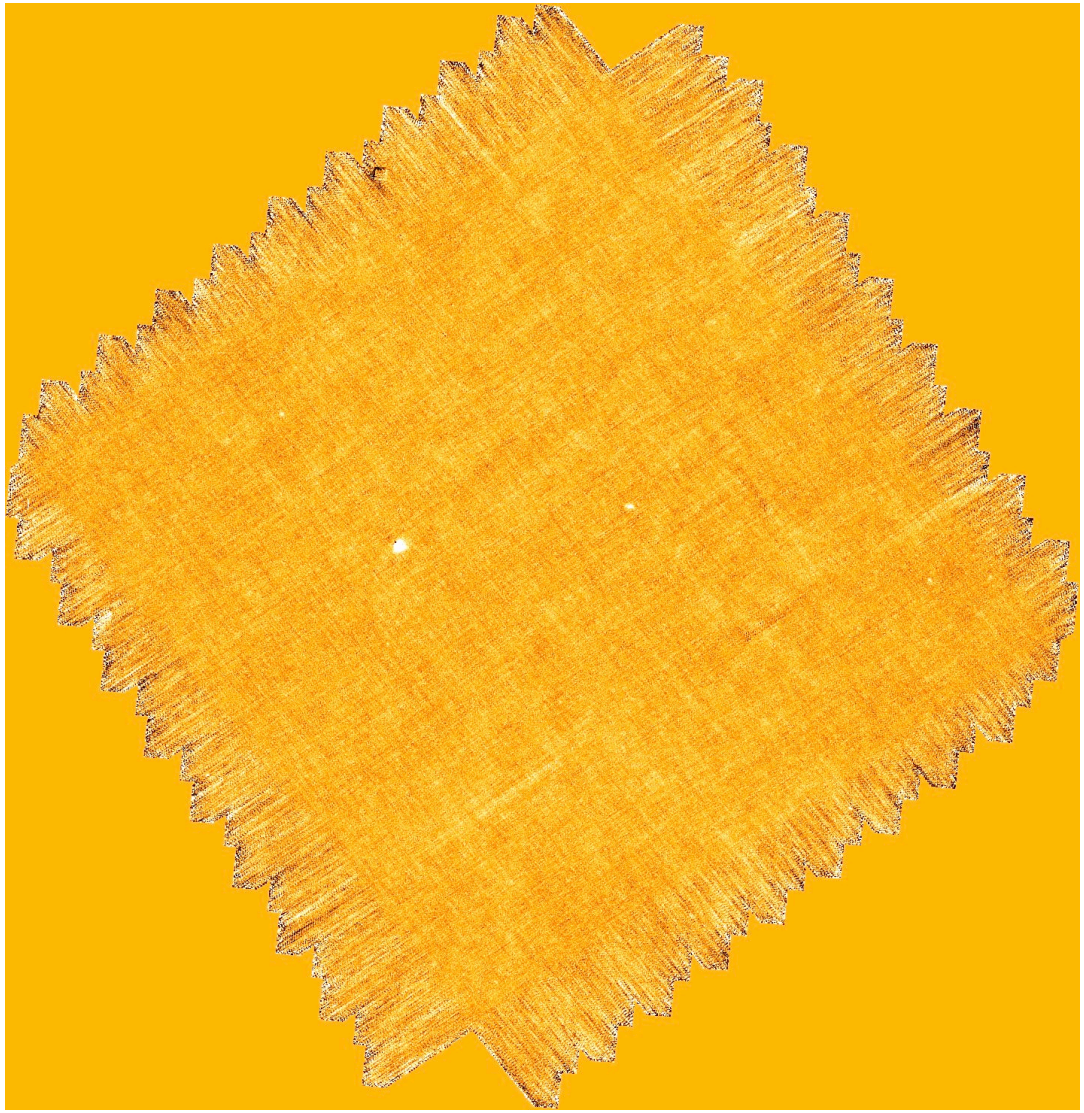




Vela



M 51 -> noisy background?



L1489 -> noisy background!