PACS Photometer - Chopping Modes:
Point-Source, Small-Source & Large Raster Mode
& Mini Scan-Map for Point-Sources and Small Fields

The PACS ICC

This document provides a brief summary of the relevant performance parameters of the chopped observing techniques in the PACS photometer and the corresponding release status:

- Point-source chopped-nodded mode (with/without dither-option): **released**
- Small-source chopped-nodded 2×2 raster mode: **cancelled**
- Large-source chopped raster mode: **cancelled**

Instead of small-source and large raster modes we recommend to use the PACS photometer scan map mode. In case of point-sources the PACS the mini scan-map option is recommended in many cases as a valid alternative with better overall performance despite the large overheads for satellite turnarounds (see section 4.1 for implementation details).

In this document we give implementation and performance details for the chop-nod and the mini scan-map technique and compared the in-orbit results with current HSpot (v4.4) predictions. We also discuss the various advantages and disadvantages of both techniques when observing point-sources.

Contents

1 Point-source chopped-nodded observations

1.1 Technical implementation: Calibration Block and On-Sky

The calibration block is executed during the target acquisition phase and lasts for about 30 s. The chopper moves with a frequency of 0.625 Hz between the two PACS internal calibration sources. 19 chopper cycles are executed, each chopper plateau lasts for 0.8 s (32 readouts on-board) producing 8 frames in the down-link. There are always 5 s idle-time between the calibration block and the on-sky part for stabilisation reasons.

The calibration blocks allow to follow the evolution of the bolometer gains during the day. There is always one such block at the beginning of each PACS photometer AOR (also in cases of grouped/concatenated observations). This single calibration block is sufficient, even for very long observations. Note: The longest possible duration of a point-source AOR in chopping-nodding techniques is 5.55 h in case of the maximum allowed repetition factor of 120.

The PACS photometer point-source mode uses the PACS chopper to move the source by about 50″, corresponding to the size of about 1 blue/green bolometer matrix or the size of about half a red matrix, with a chopper frequency of 1.25 Hz. The nodding is performed by a satellite movement of the same amplitude but perpendicular to the chopping direction. On each nod-position the chopper executes 3×25 chopper cycles. The 3 sets of chopper patterns are either on the same array positions (no dithering) or on 3 different array positions (dither option). In the dither-option the chopper pattern is displaced in ±Y-direction (along the chopper direction) by about 8.5″ (2 2/3 blue pixels or 1 1/3 red pixels). Each chopper plateau lasts for 0.4 s (16 readouts on-board) producing 4 frames per plateau in the down-link. The full 3×25 chopper cycles per nod-position are completed in less than 1 minute. The pattern is repeated on the second nod-position. In case of repetition factors larger than 1, the nod-cycles are repeated in the following way (example for 4 repetitions): nodA-nodB-nodB-nodA-nodA-nodB-nodB-nodA to minimise satellite slew times.

1.2 Sensitivity estimate

The current HSpot (v4.4) predictions in chop-nod technique are 3.0, 3.2, 4.6 mJy in blue/green/red for the instrumental 5-σ/1 h noise.

The currently achieved sensitivities are worse than the HSpot prediction. This is not due to increased noise in the instrument. The most likely causes of this are:

1. a too simple approach in the sensitivity estimate to take into account $1/f$ noise: the effective actual noise level with 1.25 Hz chopper frequency and 4 readouts per chopper plateau is higher than the adopted 3 Hz value of the 1/f noise spectrum assumed in HSpot predictions.
2. the actual optical coupling of the instrument to the sky,

3. the detector time constants which affect the first frame after each chopper transition and hence the detector responsivity. Currently, 25% of the data are discarded for that reason,
4. combination of all effects.

The noise spectrum is such that higher sensitivities would be reached for faster chopper frequencies, but due to the detector time constants the signal losses increase in a way that for a given AOR execution time the sensitivity would not improve.

Based on our current data reduction experience (see Sect. 6.1), our knowledge on detector noise behaviour in this mode and the presence of background confusion, the measured signal-to-noise is larger than HSpot predictions by a factor of at least 1.5 to 2. This affects especially the detection of the faintest sources below ∼10 mJy.

For a realistic comparison between the HSpot prediction and in-orbit performance it is necessary to take the local background confusion noise into account. The background confusion noise is in most cases not relevant in the blue band, but contributing significantly in the green band and dominating the final S/N values in the red band.

Note, that in the chop-nod technique the source is on-array all the time, but the derived differential signal corresponds to only half of the total time. The elimination of the chopper transitions cost another 25% of the time. Another 3 min are going into the calibration block and another few minutes into the satellite movements for the nodding. For a 1 hour chop-nod AOR the effectively used time is therefore roughly 20 min.

1.3 Pipeline status

The pipeline data-processing steps (apart from the obvious ones such as unit conversion and astrometry addition) are (see more details in the PACS Data Reduction Guide and Advanced User's Manual in the HIPE help):

flag known bad pixels

flag saturated pixels

convert ADUs to Volts

correct for cross-talk (currently disabled)

assign Ra/Dec to each pixel

signal averaging per chopper plateau (discard first frame of each plateau)

difference of chopped signals and cosmic rays removal

signal averaging per dither position

difference of nod-positions per dither position

combination of dither-positions

Responsivity and flat-field correction

Gain drift correction (currently disabled)

Map reconstruction

Cosmic rays removal is performed with a sigma-clipping when averaging at each dither position. Cross-talk correction (affecting mostly one column on both red matrices) is implemented, but not validated yet as we are generating the calibration file needed for the pipeline module. Responsivity, flat-field, gain drift corrections use ground-based version of the calibration product and will be updated in the course of the Science Demonstration Phase. Map reconstruction is a simple projection of the data cube after shift & add of the 4-beam chop-nod pattern.

The final map covers about $2.5' \times 4.5'$, but with uneven coverage and beam pattern. Only the central $\sim 50'' \times 50''$ receive full coverage and correct addition of beams of the targeted source, other regions can help diagnosing contamination by other sources in the field.

The latest script to reduce chop-nod point-source observations is given in Sect. 6.1.

2 General photometer aspects

2.1 Point spread function

The photometer PSF is characterised by:

- A narrow core which is round in the blue bands but slightly elongated in spacecraft Z direction in red.
- A tri-lobe pattern seen at the several % level in all bands, most clearly in the blue with its strongest signal, and ascribed to imperfect mirror shape.
- Knotty structure at sub-percent level, clearly seen in blue and indicated in green.

An illustration of the PSF is shown in Figure 1.

These PSFs and derived quantities reflect the intrinsic optical quality of Herschel+PACS. In the chop-nod point-source reduction they will be slightly smeared, in particular at short wavelengths, according to telescope pointing jitter and drifts.

2.2 Flux calibration

The flux calibration of the PACS photometer data is a three-stage process: the data have to be flat-fielded, engineering units are converted to Jy/pixel (the responsivity correction), and pixel gains are corrected for with respect to a reference consistent with both the flat-field and the responsivity calibration products (to account for small gain drifts with time).

Based on a large set of celestial flux calibrators ranging from fluxes below 50 mJy to several hundred Jansky, the flux calibration is very reliable. The estimated **absolute flux accuracy** is better than 10% in the blue and green bands and better than 20% in the red band.

However using a set 5 primary PACS calibration standard stars (α Boo, α Tau, α CMa, α Ceti & γ Dra,) in the flux range 0.6-15 Jy in the 3 bands, the absolute flux accuracy is found to be better than 5% in the blue and green and better than 10% in the red.

The aperture correction factors can be taken from Fig. 2 (PICC-ME-TN-033, Version 0.3) or from the tables in Sect. 7.

The flux calibration of the PACS photometer rests on :

1. a spectral convention of $\nu \times f_\nu =$ constant and reference filter wavelengths at 70, 100 and 160 μm.
2. a set of celestial flux calibrators that are mostly stars and asteroids for which model spectra are available that allow either color-correction of the measured spectral densities in order to compare them with the predicted monochromatic model fluxes or computation of the expected measured spectral densities.

NOTE: In the current public HIPE versions (v2.x.x) there is still an error in the flux scale which has to be corrected manually. This error in the flux scale is related to the responsivity calibration file version 3. The final reduced fluxes, after aperture- and colour-correction, are too high and have to be scaled down by factors of 1.05 in the blue band, 1.09 in the green band, and 1.29 in the red band. This flux overestimation will be corrected in the next released valid calibration files as part of HIPE release 3.0.

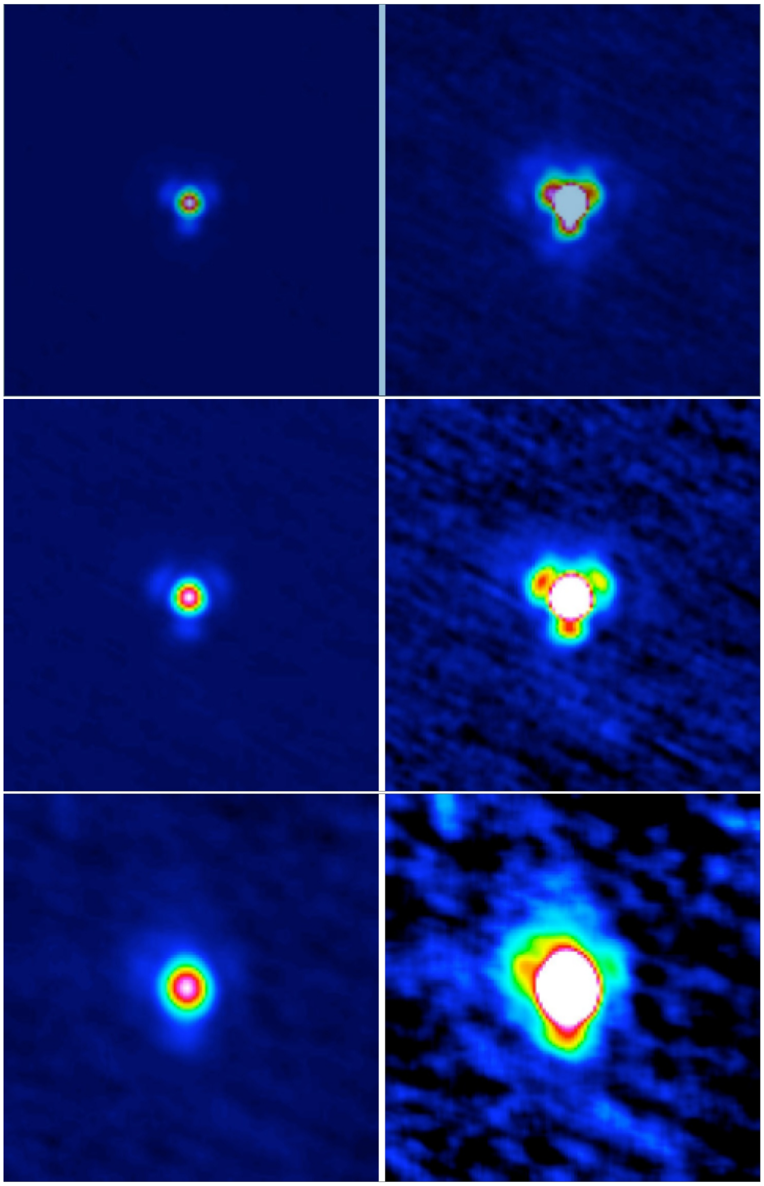


Figure 1: The photometer PSF in blue, green and red (top to bottom) derived from scans performed at 10″/s. Left-hand panels display the image with a linear scale up to the peak, while right-hand panels show up to 10% of the peak.

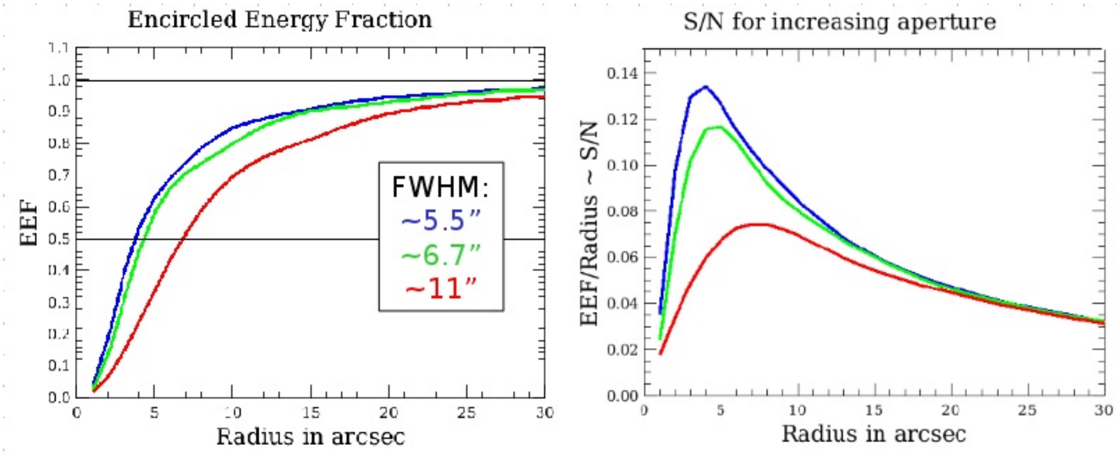


Figure 2: Left: Encircled energy fraction as a function of circular aperture radius for the three bands. Derived from slow scan OD160 Vesta data. The EEF fraction shown is normalized to the signal in aperture radius 60arcsec, with background subtraction done in an annulus between radius 61 and 70 arcsec. The right panel shows the corresponding S/N curve under the assumption that noise scales linearly with aperture radius. Note that this assumption is not met for scanmaps with 1/f noise.

2.3 Pointing

The Relative Pointing Error (RPE) is about 0.3″ implying a high stability for the chopping phases. Due to the Spatial Relative Pointing Error (SRPE) of about 1.5″ a slight broadening of the PACS PSF is expected when combining the images of the two nods, in particular in the blue/green bands where the PSF is sharper. The Absolute Pointing Error (APE), which affects the complete chop-nod pattern, is better than 2″.

2.4 Digitization

Because of satellite data-rate issues, we have had to add a supplementary compression stage called bit-rounding before the data is down-linked. This means that while we average 4 images on board, we also round the last n bits of the result. The default value for n was 2 during the Performance Validation campaign (high-gain observations performed with bit-rounding of 2 are effectively digitized with a step of $2\,10^{-5}$ V, or 4 ADU), now released to 1 bit only since the Nov. 2, 2009. The PACS ICC is currently experimenting with dispensing with this compression step in some observing modes, e.g. using no bit rounding during scan-map mode, but it is not clear if the increased datarate could then cause problems in long ODs where also the entire Daily Tele-Communications Periods (DTCP) is used for science observations (which is possible now in Routine Phase). A final decision will be taken in spring 2010 when all the satellite, instrument and down-link constraints are known.

3 Discussion on the point-source mode

3.1 Caveats of chop-nod Point-Source mode

The biggest caveat on the chop-nod technique is the sensitivity, which is lower than the HSpot prediction. All science programmes planning to use this mode on faint point-sources should consider to switch to the more sensitive mini scan-map mode (Sect. 4).

The implemented chop-nod technique has also limitations in case of nearby sources (all 3 bands) and confusing background structures (mainly in the red band). In these cases the various chopped-nodded beams might overlap in the final map and accurate photometry is more difficult (see Fig. 3, left side).

The area of homogeneous coverage in the final image is limited to a region of side length 8-10 pixels around the source. Outside this region the noise properties are different due to a lower pixel coverage (see Fig. 3, right side).

The chop-nod technique relies strongly on the performance of very few pixels. Some of these key pixels are meanwhile known to show intermittent variability (column 11 in blue matrix 6), i.e. with potentially higher noise levels. It is not possible to completely avoid the bad pixels due to array size constraints, possible astrometric uncertainties for the science targets and the APE of the satellite.

The PACS chop-nod technique is not suitable for the follow-on option which was foreseen for moving solar system targets to catch the object a second time after it has moved by about 3 FWHM. The final map is too much disturbed by the various beams and the poorer pixel coverage at these distances from the map centre does not allow to fully characterise the background as it was planned via the follow-on option.

3.2 Advantages of chop-nod Point-Source mode

One advantage of the chop-nod technique is the stability of the reconstructed PSF due to the small RPE of 0.3″.

The chop-nod technique (in combination with the dither-option) provides high photometric accuracy for isolated sources. The data-reduction script (Sec. 6.1) is consolidated and produces photoemetrically reliable maps.

Many stellar calibration sources have meanwhile been observed in the described chopping-nodding technique during PV phase, allowing to reduce and calibrate the science target observations in a very controlled and reliable way.

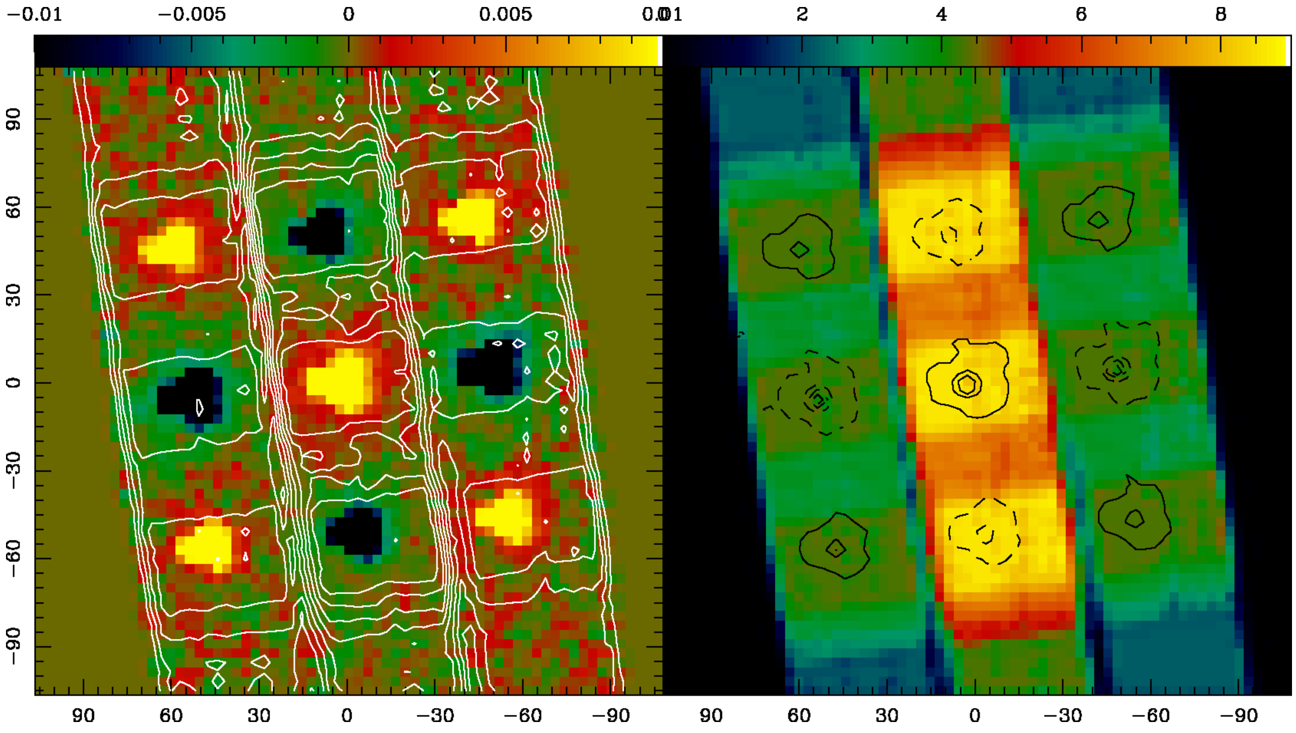


Figure 3: Synopsis of the final map (left) and the pixel coverage (right) after the final processing step with all chop/dither/nod positions projected onto the sky for a bright target with several Jansky. It demonstrates the limitation in aperture photometry to find clean background regions due to the various positive and negative beams. The coverage map illustrates that only a fraction of the central image of the target has the full pixel coverage.

The chop-nod mode allows to obtain very high spatial resolution which might be needed for resolving small-scale structures. But since there is no ”cluster pointing-mode” available, such a sub-pixel (satellite) dithering is only possible by entering serveral concatenated/grouped AORs with slightly shifted target positions.

3.3 Chop-nod mode implementation for point-sources

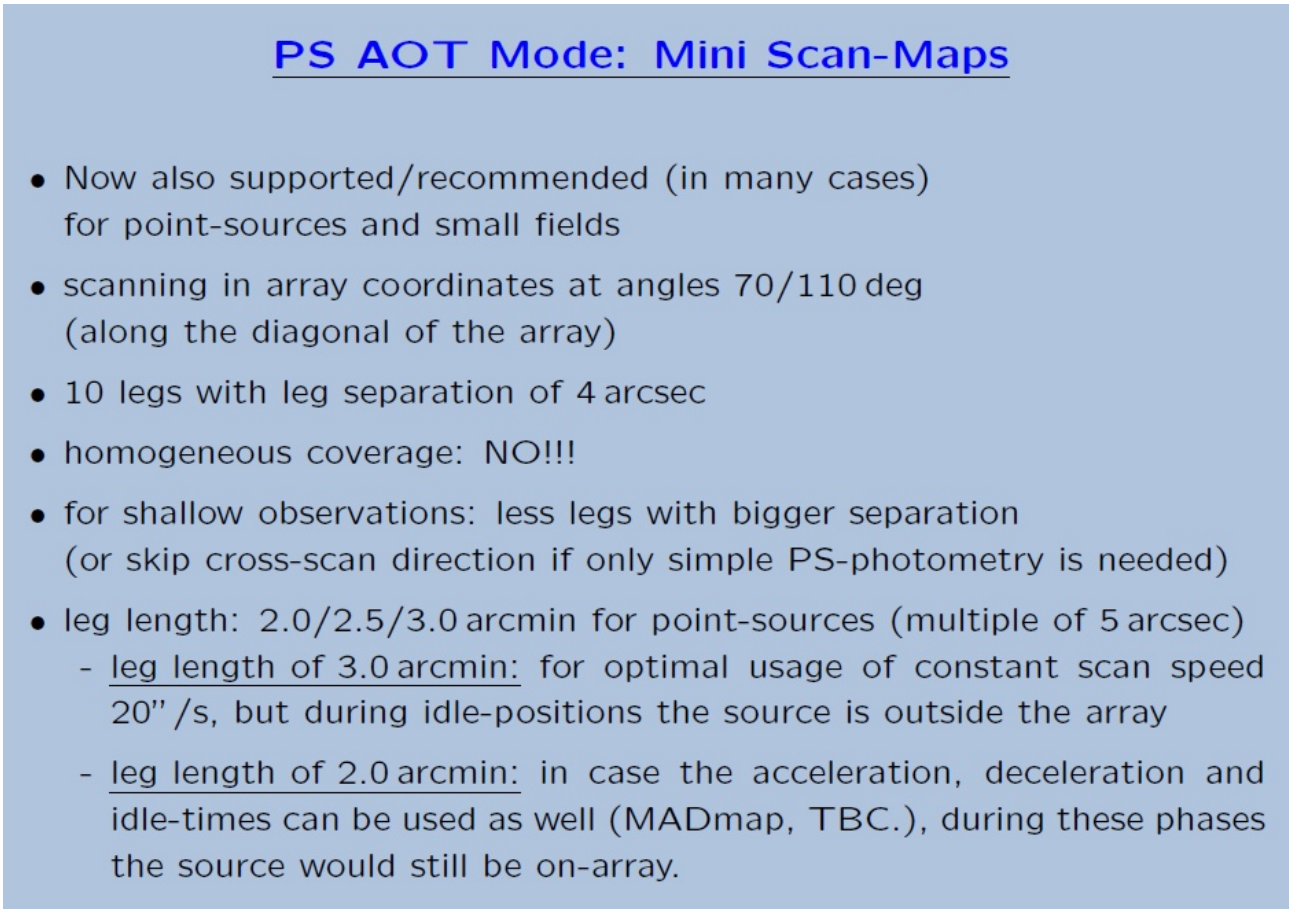
If the the chop-nod point-source mode is used, it is recommended to use the dither-option to improve the PSF-sampling for fainter targets. However it is reminded that the mini scan map mode is superior in terms of sensitivity for the faintest sources.

Due to symmetry reasons in the analysis of the nod-positions we recommend to use either a repetition factor of 1 (nodA - nodB) or a multiple of 2 for the repetition factor (example for a repetition factor of 4: nodA-nodB-nodB-nodA - nodA-nodB-nodB-nodA). In the analysis, the data are sliced by nod-blocks consisting of ”nodA-nodB-nodB-nodA” and later a recombination of these slices images is performed

4 PACS scan map mode for point-sources

Alternatively it is possible to use the PACS scan map mode also for point-sources. Despite the large overheads for the satellite turnarounds between the short scans, this option has a better performance with respect to point-source S/N within the same AOR execution times.

4.1 Scan map mode implementation for point-sources



PS AOT Mode: Mini Scan-Maps

- Now also supported/recommended (in many cases) for point-sources and small fields
- scanning in array coordinates at angles 70/110 deg (along the diagonal of the array)
- 10 legs with leg separation of 4 arcsec
- homogeneous coverage: NO!!!
- for shallow observations: less legs with bigger separation (or skip cross-scan direction if only simple PS-photometry is needed)
- leg length: 2.0/2.5/3.0 arcmin for point-sources (multiple of 5 arcsec)
  - leg length of 3.0 arcmin: for optimal usage of constant scan speed 20"/s, but during idle-positions the source is outside the array
  - leg length of 2.0 arcmin: in case the acceleration, deceleration and idle-times can be used as well (MADmap, TBC.), during these phases the source would still be on-array.

Figure 4: Summary of the recommendations for mini scan-map implementation for point-sources.

In case of using the scan map mode for point-sources and very small fields we propose the following configuration:

- medium scan speed ($20''/s$)
- scan angle in array coordinates along the array diagonal: 110° or 70°
  (in case of cross-scans: 110° and 70°)
- scan lengths: 2.0, 2.5, 3.0, 3.5 or $4.0'$ (the array diagonal has about $4.0'$), the scan leg lengths should be a multiple of $5''$
- small and even number of scans: 4, 6, 8, 10, ... (for minimisation of satellite movements and a match to the array diagonal)

- small leg separation: 2...5″ with the smaller separation for a larger number of scan legs and vice versa (to have the source on-array in all legs). Examples: 8 scan legs of 3′ lengths and 4″ separation. This map would then match a sky region with the width of about 3×FWHM$_{red}$ with very high coverage.
- repetition factor: as needed to reach the required sensitivity
- cross-scan maps: would allow to apply all kinds of map-making techniques and not just the high-pass filtering. Probably also required for higher quality photometry and better spatial characterisation in the near source vicinity. It is recommended to group/concatenate the 2 cross-scan AORs to minimise slew overheads. Each AOR will have its own 30 s calibration block.

Note: Part of the satellite turnarounds are still containing useful information: For the $20''/s$ scan speed it is possible to use, in addition to the scan legs, about 0.8 s (i.e., 16″) prior to each scan leg and about 1.5 s (i.e., 30″) after the official scan leg has finished. During these times the scan speed is already/still very close to the requested value.

Fig. 4 summarises the recommendations for mini scan-map implementation for point-sources. Fig. 5 shows the simulated movement of the source for the recommended 10-leg mini scan-map on the array and the corresponding coverage map as produced by HSpot.

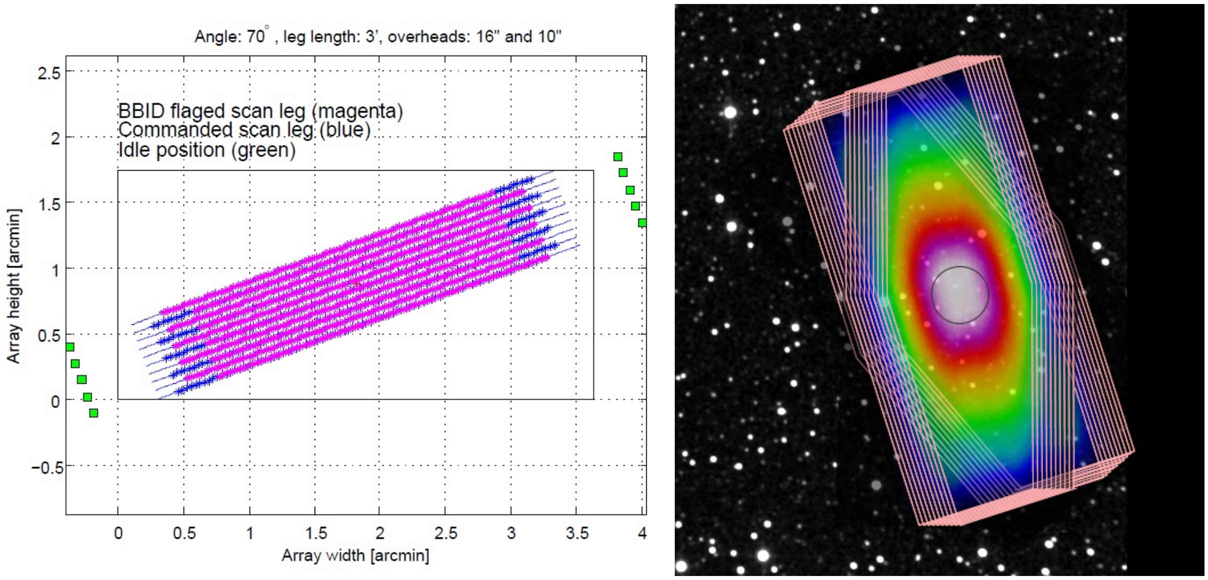


Figure 5: Left: simulated source movement on-array for 3.0 arcmin scan leg length, including the useful leg extensions which are still at constant speed (and idle-positions). Right: Coverage map for 2 mini scan-maps at array angles of 70 and 110 deg. The homogeneous, high-coverage area (circle) is about 50″ in diameter.

4.2 Mini Scan-Maps: Performance

- homogeneous high coverage area of ∼50 arcsec diameter
- on-source leg times (useful times at ∼20″/s speed):
  2.0 arcmin: (6.0 -1.0)+0.8+1.5 s = 7.3 s
  2.5 arcmin: (7.5 -1.0)+0.8+1.5 s = 8.8 s
  3.0 arcmin: (9.0 -1.0)+0.8+1.5 s = 10.3 s
- On-source time, AOR execution time, efficiency (2 AORs, 2-bands):

		2.0 arcmin			2.5 arcmin			3.0 arcmin	
	OST	concat. AORs	Eff.	OST	concat. AORs	Eff.	OST	concat. AORs	Eff.
1 rep	146	426+ 305 s	0.20	176 s	446+ 325 s	0.23	206 s	456+ 335 s	0.26
2 rep	292	678+ 557 s	0.24	352 s	718+ 597 s	0.27	412 s	738+ 617 s	0.30
3 rep	438	930+ 809 s	0.25	528 s	990+ 869 s	0.28	618 s	1020+ 899 s	0.32
5 rep	730	1434+ 1313 s	0.27	880 s	1534+1413 s	0.30	1030 s	1584+1463 s	0.34
6 rep	876	1686+ 1565 s	0.27	1056 s	1806+1685 s	0.30	1236 s	1866+1745 s	0.34

- Instrumental 1-σ rms-noise (blue/green/red): 27/29/41 mJy / $\sqrt{t_{total}}$

	2.0 arcmin	2.5 arcmin	3.0 arcmin
1 rep	2.2/2.4/3.4 mJy	2.0/2.2/3.1 mJy	1.9/2.0/2.9 mJy
2 rep	1.6/1.7/2.4 mJy	1.4/1.5/2.2 mJy	1.3/1.4/2.0 mJy
3 rep	1.3/1.4/2.0 mJy	1.2/1.3/1.8 mJy	1.1/1.2/1.6 mJy
5 rep	1.0/1.1/1.5 mJy	0.9/1.0/1.4 mJy	0.8/0.9/1.3 mJy
6 rep	0.9/1.0/1.4 mJy	0.8/0.9/1.3 mJy	0.8/0.8/1.2 mJy

- 1-σ confusion noise levels for PACS point-sources
  in blue/green/red: ≥ 0.03 mJy, ≥ 0.5 mJy & ≥ 1.8 mJy
- derived/measured 5-σ/1 h noise for PACS (3.0 arcmin legs):

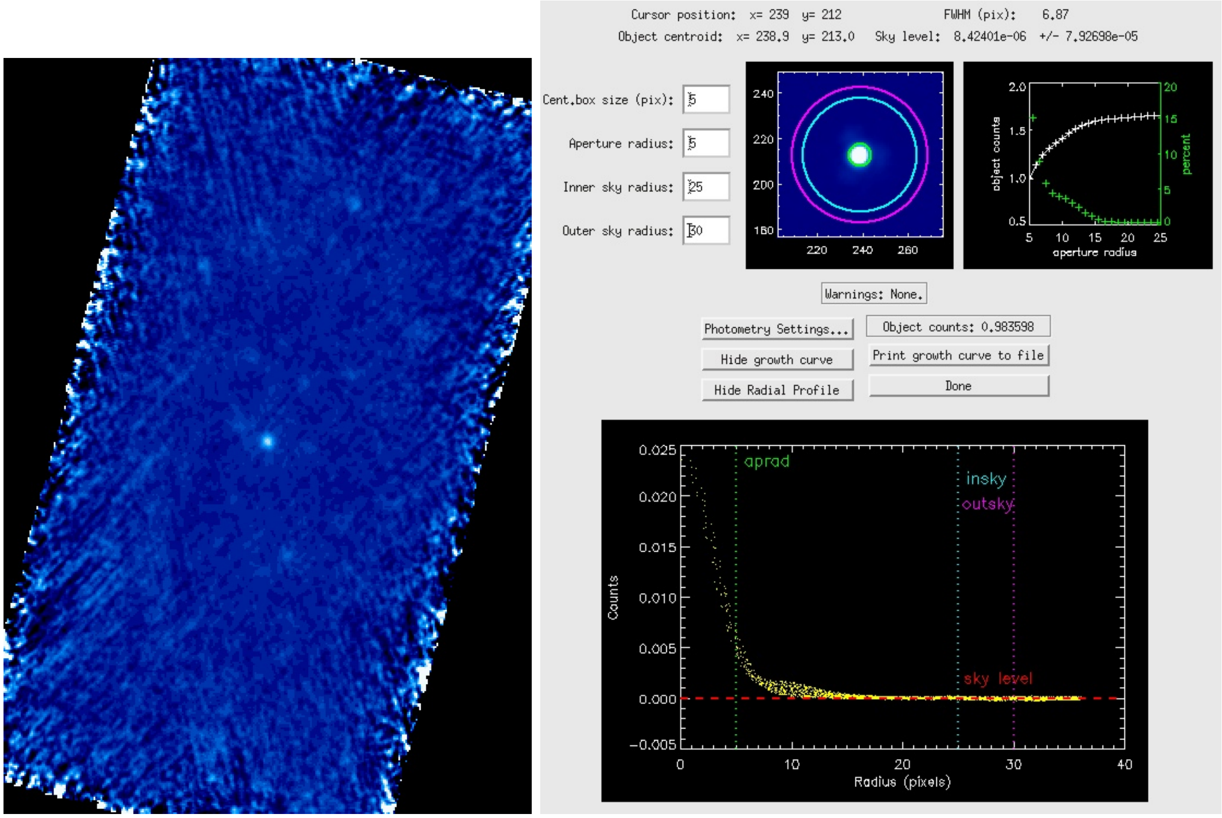
	instrumental	bgr. conf.		total 5-σ	Note
blue	0.77 mJy	$\geq$ 0.03 mJy	×5	**$\geq$3.9 mJy**	instr. limited
green	0.82 mJy	$\geq$ 0.5 mJy	×5	**$\geq$4.8 mJy**	instr./confN limited
red	1.17 mJy	$\geq$ 1.8 mJy	×5	**$\geq$10.7 mJy**	confN limited

- Current HSpot (4.4) values for instrumental 1-σ/1 h
  (min-scan map: 10 legs with 3.0 arcmin length, separation 4″):
  1.5 mJy, 1.6 mJy, 2.2 mJy in blue, green, red band
- small aperture are sometimes problematic (slightly deformed PSFs)
  10-15″ in blue/green and 20″ in red work nicely
- sky should be taken at least 30-35″ distance from source (in blue/green) and 40-50″ distance from source in red

4.3 Advantages of scan maps for point-sources

The advantages for the scan map mode are:

- it provides a better characterisation of the close vicinity of the target and larger scale structures in the background
- also targets with positional uncertainties of 10″ or more are still perfectly covered
- the final map has a much larger area of homogeneous coverage (about 50″ in diameter) depending on observation configuration, see Fig. 5)
- more pixels see the target, the impact of noisy, variable and dead pixels is less problematic
- better point-source sensitivity in all bands as a high-pass filter can be used to remove 1/f noise up to higher frequencies.



Cursor position: x= 239 y= 212 FWHM (pix): 6.87
Object centroid: x= 238.9 y= 213.0 Sky level: 8.42401e-06 +/- 7.92698e-05

Cent.box size (pix): 5
Aperture radius: 5
Inner sky radius: 25
Outer sky radius: 30

Warnings: None.

Object counts: 0.983598

Figure 6: Left: combined cross-scan mini-maps for HD 159330 (∼30 mJy at 100 μm.) Right: Growth curve and radial profile for a combined cross-scan mini-map on γ Dra.

- no negative beam in final map.

4.4 Disadvantages of scan maps for point-sources

The small array size requires the implementation of short scan legs of 2 to 4′ only. Combined with the optimum scan speed of $20''/s$ this leads to short on-scan times of less than 12 s. The corresponding satellite turnarounds between these scan legs take currently about 20 s. It is expected that the large overheads for satellite turnover loops can still be reduced by a few percent (instead of 5 s idle-time only 2-3 s idle-time), but the timescales for this change is not known. But the ratio between useful on-array time and satellite manoeuvre times is still low.

4.5 Direct comparision of both modes

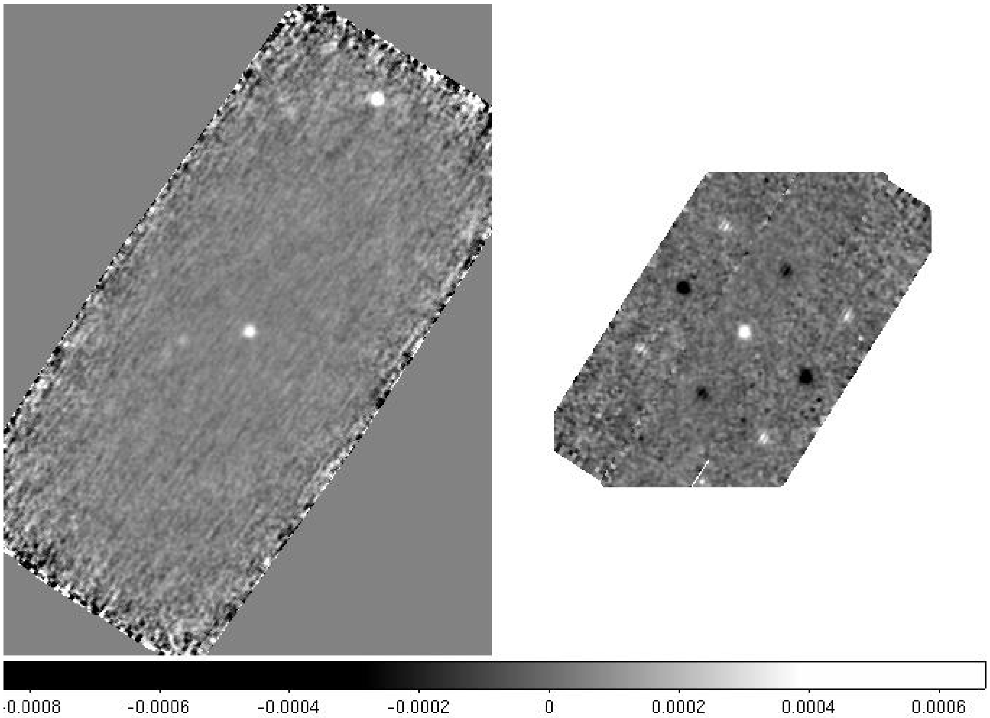


Figure 7: HD 1382865 (OD132) observed in scan-map mode (left) and chop-nod (right) for a similar AOR execution time in the blue band (70 μm), displayed with the same image cut levels.

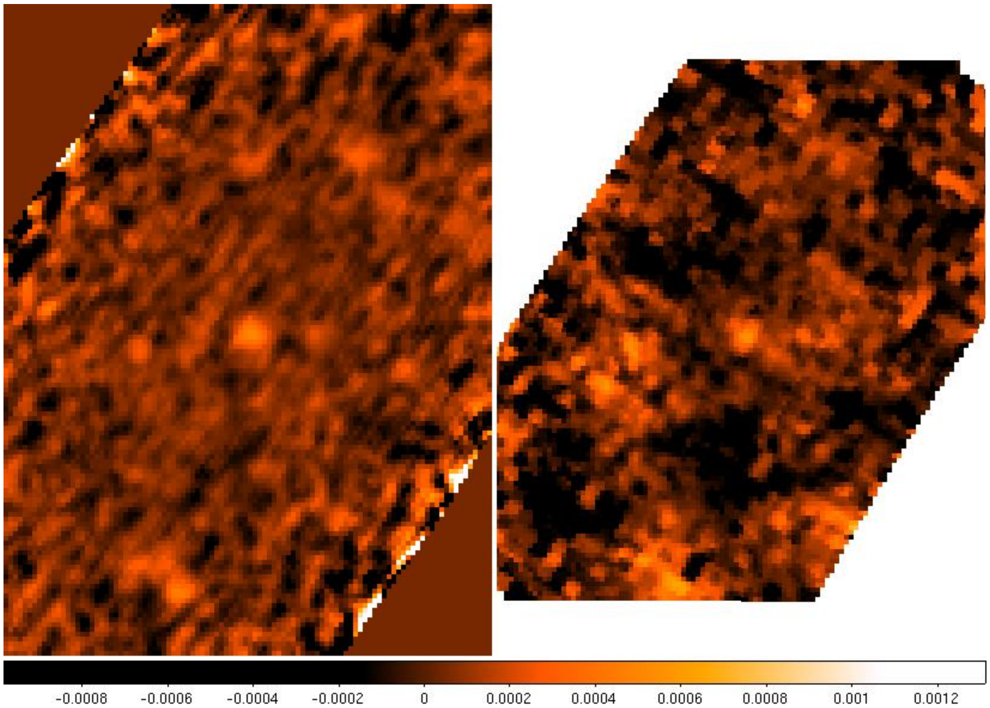


Figure 8: HD 1382865 (OD132) observed in scan-map mode (left) and chop-nod (right) for a similar AOR execution time in the red band (160 μm), displayed with the same image cut levels.

5 Small-source Mode and Large Raster Mode

Both modes can be replaced by the Scan Map Mode implementation without loosing sensitivity performance. But for very small fields it is advisable to wait for the final optimisation of the satellite turnaround loops. A significant increase of on-source is expected when the idle-times during the turnaround loops are minimised. It is planned to have this change available towards the end of November 2009.

With this optimisation in the Scan Map Mode we consider the chopped-nodded Small Source Mode as obsolete. Already implemented observations should be moved to the Scan Map Mode following the implementation recommendations in Section 4.1. Nevertheless, several reference measurements in this 2×2 raster fields with off-array chopping and nodding are available as part of the photometer validation programme.

The Large Raster Mode, with the availability of a large range of possible steps and step sizes, has been executed during performance verification phase in different flavours. But in all considered cases the repetition of these fields in Scan Map Mode would provide more reliable products and in most cases also better sensitivity. It is currently not foreseen to commission this mode.

6 Data reduction scripts

6.1 Data reduction scripts: chop-nod technique for point-sources

This script has also been used to analyse the large number of calibration observations on celestial standards taken in this mode. It is considered by the PACS ICC as a very consolidated reduction procedure with high photometric reliability.

```
#########################################
# PACS PHOT PS AOT Script with slicing
# 19 February 2010
# Markus Nielbock (MPIA Heidelberg)
#########################################
from herschel.share.unit import *
from herschel.pacs.signal.context import PacsContext
from herschel.pacs.spg            import PacsSliceContextTask
from herschel.pacs.spg.pipeline import *
from herschel.pacs.signal.context import *
from herschel.pacs.signal import SlicedFrames

import os
dir = ’/home/username/MyDirectoryPath’ $ change username/MyDirectoryPath to directory path where you want
# dir = os.getcwd()+"/"   # work on Linux only on some HIPE versions only ...

# To be modified to match observation
POOLDIR=’/home/username/.hcss/lstore/’
POOLNAME=’OSBID1342186141_PS_AOT’
OBSID=1342186141
OD=160
AOR="PVPhotAOTVal_511J_StdPS_2rep_blu_HD148387_001"
TARGET="HD148387"

# select blue or red camera
camera = ’blue’

# Retrieve the necessary data
print "Retrieving observation context ..."
obs=getObservation(long(OBSID),poolName=POOLNAME, poolLocation=POOLDIR,verbose=False)
# Alternative version for MPE grid:
#obs=getObservation(long(OBSID), od=OD, poolLocation=POOLDIR,verbose=True)

# the metatdata do not contain all information for some OBSIDs
if (obs.meta.containsKey("object")):
    print "Reading target name from metadata ..."
    object=obs.meta["object"].value
    TARGET=object.replace(" ","")
#
if (obs.meta.containsKey("aorLabel")):
    print "Reading AOR label from metadata ..."
    AOR=obs.meta["aorLabel"].value
#
if (obs.meta.containsKey("odNumber")):
    print "Reading OD number from metadata ..."
```

```
    OD=obs.meta["odNumber"].value

# retrieve auxiliary products:
# pointing product
print "Extracting pointing product ..."
pp = obs.auxiliary.pointing

# orbit ephemeris
print "Extracting OrbitEphemeris ..."
oep = obs.auxiliary.orbitEphemeris
horizons = None

# Is it a solar System Object ?
isSso = False
if (obs.meta.containsKey("naifid")):
 if (obs.meta["naifid"].value != 0):
   isSso  = True

# instrument calibration
print "Extracting calibration database ..."
calTree=getCalTree()

# time correlation
print "Extracting Time Correlation product ..."
timeCorr = obs.auxiliary.timeCorrelation

# instrument housekeeping
print "Extracting photometric housekeeping product ..."
photHK=obs.level0.refs["HPPHK"].product.refs[0].product["HPPHKS"]

# get the Level 0.5 data cube (frames) and access either the blue or red channel frames
# data are sliced in packets of one ABBA nodding cycle
print "Starting processing from level 0.5 ..."
level0_5 = PacsContext(obs.level0_5)
slicedFrames  = level0_5.averaged.getCamera(camera).product
pacsPropagateMetaKeywords(obs,’0.5’,slicedFrames)

# select pixel size of final map and correct wavelength information
if (camera == ’blue’):
    pixsize=1.0
    if (slicedFrames.refs[1].product.meta["blue"].getValue() == "blue1"):
       filter = ’blue’
       lam = 70.0
    else:
       filter = ’green’
       lam = 100.0
elif (camera == ’red’):
    pixsize=2.0
    filter = ’red’
    lam = 160.0

# what is the size of the cube I want to process ?
print "Data cube dimension: "+str(slicedFrames.refs[1].product.signal.dimensions)
```

```
# the following lines are advised to correct filter information for old (pre OD150?) L0 processed data.
# the demo data are from OD 160, so this part is not needed
if camera == 'blue':
  # calibration block slice
  wpr=slicedFrames.refs[0].product.getStatus("WPR")
  band=slicedFrames.refs[0].product.getStatus("BAND")
  if wpr.where(wpr == 0).length() > 0:
    if band[wpr.where(wpr == 0)][0]=='BS':
      print 'WARNING for blue filter: WPR=0 was erroneously assigned BS, now reset to BL'
      band[wpr.where(wpr == 0)] = String('BL')
  if wpr.where(wpr == 1).length() > 0:
    if band[wpr.where(wpr == 1)][0]=='BL':
      print 'WARNING for blue filter: WPR=1 was erroneously assigned BL, now reset to BS'
      band[wpr.where(wpr == 1)] = String('BS')
  slicedFrames.refs[0].product.setStatus("BAND",band)
  # science block slice
  wpr=slicedFrames.refs[1].product.getStatus("WPR")
  band=slicedFrames.refs[1].product.getStatus("BAND")
  if wpr.where(wpr == 0).length() > 0:
    if band[wpr.where(wpr == 0)][0]=='BS':
      print 'WARNING for blue filter: WPR=0 was erroneously assigned BS, now reset to BL'
      band[wpr.where(wpr == 0)] = String('BL')
  if wpr.where(wpr == 1).length() > 0:
    if band[wpr.where(wpr == 1)][0]=='BL':
      print 'WARNING for blue filter: WPR=1 was erroneously assigned BL, now reset to BS'
      band[wpr.where(wpr == 1)] = String('BS')
  slicedFrames.refs[1].product.setStatus("BAND",band)

# get the number of sliced nod cycles in observation :
noofsciframes=slicedFrames.numberOfScienceFrames

# run the pipeline for each nod cycle
for i in range(noofsciframes):
    # ++++++++++++++++++++++++++++++++++++++++++++++++
    # Extract one slice and combine it with the calibration block
    # ++++++++++++++++++++++++++++++++++++++++++++++++
    print 'Processing nod cycle '+str(i+1)+' of '+str(noofsciframes)+' ...'
    # calibration block has index 0
    framesnod = slicedFrames.getCal(0).copy()
    # retrieve the science data per nod cycle
    sciframesnod = slicedFrames.getScience(i).copy()
    # merge both slices to one dataset
    framesnod.join(sciframesnod)
    del(sciframesnod)
    #
    # ++++++++++++++++++++++++++++++++++++++++++++++++
    # Processing
    # ++++++++++++++++++++++++++++++++++++++++++++++++
    print "Extracting, processing and removing calibration block ..."
    framesnod = detectCalibrationBlock(framesnod)
    calBlock = photCSExtraction(framesnod)
    calBlock = photCSProcessing(calBlock,photHK,calTree)
    framesnod = photCSClean(framesnod,calBlock)
    framesnod = removeCalBlocks(framesnod)
```

```
print "Flagging bad pixels ..."
framesnod = photFlagBadPixels(framesnod,calTree=calTree)
# flag additional bad pixel
if (camera == ’blue’):
    blue_badpix=calTree.photometer.badPixelMask.blue
    blue_badpix[2,30]=1
    framesnod.setMask("BADPIXELS",blue_badpix)
print "Flagging possible saturation ..."
framesnod = photFlagSaturation(framesnod,calTree=calTree,hkdata=photHK,check=’full’)
print "Converting digital values to volts ..."
framesnod = photConvDigit2Volts(framesnod,calTree=calTree)
# ground-based correction is overcorrecting, hence switched off for the time being
#framesnod = photCorrectCrosstalk(framesnod)
print "Adding UTC reference time frame ..."
framesnod = addUtc(framesnod, timeCorr)
print "Converting chopper readout to angles ..."
framesnod = convertChopper2Angle(framesnod,calTree=calTree)
print "Adding coordinates ..."
framesnod = photAddInstantPointing(framesnod,pp, calTree=calTree, orbitEphem=oep, horizons=horizons, i
print "Masking frames affected by chopper movement ..."
framesnod = cleanPlateauFrames(framesnod)
print "Determining readout frames per dither position ..."
framesnod = photMakeDithPos(framesnod)
print "Identifying raster positions ..."
framesnod = photMakeRasPosCount(framesnod)
print "Averaging detector readouts per chopper position ..."
framesnod = photAvgPlateau(framesnod,skipFirst=True,copy=1)
print "Correct pointing information ..."
framesnod = photAddPointings4PointSource(framesnod)
framesnod = photAssignRaDec(framesnod,calTree=calTree)
print "Producing differentiated chopped signal ..."
framesnod = photDiffChop(framesnod)
print "Averaging detector readouts per dither position ..."
framesnod = photAvgDith(framesnod,sigclip=3.)
print "Subtracting nodded signal ..."
framesnod = photDiffNod(framesnod)
print "Averaging detector readouts per nod position ... ..."
framesnod = photCombineNod(framesnod)
print "Correcting detector flatfield and calibrate flux ..."
framesnod = photRespFlatfieldCorrection(framesnod,calTree=calTree)
#framesnod = photDriftCorrection(framesnod,calTree=calTree)
#
################################################
# Level 1 -> Level 2
################################################
#
# project all images onto a map with a predefined pixel size, a different pixel size can be used.
print "Producing map ..."
map = photProjectPointSource(framesnod,allInOne=1,calTree=calTree,outputPixelsize=pixsize,calibration=
# include and correct available metadata
for m in(obs.meta.keySet()):
    map.meta[m]=obs.meta[m]
pass
map.meta["wavelength"].setValue(lam)
```

```
    # save maps (data cubes) into fits files
    # the FITS header is produced from the metadata
    print "Saving maps to FITS files ..."
    product = simpleFitsWriter(map,dir+'/PSmap_'+str(OBSID)+'_'+filter+'_nod'+str(i)+'.fits')
    #del(framesnod)
pass


###############################################
# Combine all NOD cycles if there are more than one
###############################################

if (noofsciframes > 1):
    from java.util import ArrayList
    from herschel.ia.toolbox.image import MosaicTask

    # making an empty list in which we are going to store the images
    images=ArrayList()
    print "Combining sliced data to master maps ..."
    for i in range(slicedFrames.numberOfScienceFrames):
      ima = simpleFitsReader(dir+'/PSmap_'+str(OBSID)+'_'+filter+'_nod'+str(i)+'.fits')
      # this swap is performed because the exposure map is empty
      # needed for weighting maps
      ima.exposure = ima.coverage
      images.add(ima)

    # mosaicking
    mosaic=MosaicTask()(images=images,oversample=0)
    Display(mosaic)

    # save final image in a FITS file :
    simpleFitsWriter(mosaic,dir+"/PSmosaic_map_"+str(OBSID)+"_"+filter+".fits")
else:
    mosaic=map.copy()

# clean up memory
System.gc()
```

6.2 Data reduction scripts: mini-scan maps for point-sources

TBD.

7 Tables for the encircled energy function

Table 1: Encircled energy fraction as a function of circular aperture radius for the three bands. Derived from slow scan OD160 Vesta data.The EEF fraction shown is normalized to the signal in aperture radius 60 arcsec, with background subtraction done in an annulus between radius 61 and 70 arcsec.

	encircled energy fraction			Radius	encircled energy fraction		
Radius [″]	blue	green	red	[″]	blue	green	red
1	0.047	0.032	0.018	31	0.978	0.978	0.956
2	0.214	0.156	0.069	32	0.979	0.980	0.959
3	0.402	0.318	0.146	33	0.981	0.981	0.963
4	0.548	0.474	0.241	34	0.982	0.983	0.966
5	0.642	0.595	0.341	35	0.983	0.984	0.969
6	0.701	0.672	0.438	36	0.984	0.985	0.972
7	0.750	0.718	0.524	37	0.985	0.986	0.975
8	0.794	0.749	0.597	38	0.986	0.987	0.977
9	0.830	0.778	0.656	39	0.987	0.988	0.980
10	0.856	0.809	0.700	40	0.988	0.989	0.982
11	0.873	0.840	0.734	41	0.989	0.990	0.983
12	0.886	0.866	0.759	42	0.989	0.991	0.985
13	0.895	0.885	0.781	43	0.990	0.992	0.987
14	0.904	0.900	0.801	44	0.991	0.993	0.988
15	0.913	0.910	0.820	45	0.992	0.994	0.990
16	0.922	0.917	0.838	46	0.992	0.994	0.991
17	0.931	0.923	0.855	47	0.993	0.995	0.992
18	0.938	0.928	0.871	48	0.993	0.996	0.993
19	0.945	0.932	0.885	49	0.994	0.996	0.994
20	0.949	0.938	0.897	50	0.995	0.997	0.995
21	0.953	0.943	0.907	51	0.995	0.997	0.996
22	0.957	0.948	0.916	52	0.996	0.997	0.997
23	0.960	0.954	0.923	53	0.997	0.998	0.998
24	0.963	0.958	0.929	54	0.997	0.998	0.998
25	0.966	0.963	0.934	55	0.998	0.998	0.999
26	0.968	0.966	0.938	56	0.998	0.999	0.999
27	0.970	0.970	0.942	57	0.999	0.999	0.999
28	0.973	0.972	0.946	58	0.999	0.999	0.999
29	0.974	0.975	0.949	59	1.000	1.000	1.000
30	0.976	0.977	0.953	60	1.000	1.000	1.000