

pnCCD processing pipeline

Content

- [Content](#)
- [pnCCD overview](#)
- [Data for tests](#)
- [Psana modules for pnCCD](#)
 - [Module `ImgAlgos.PncddNDArrProducer`](#)
 - [Modified module `ImgAlgos.PncddImageProducer`](#)
 - [Sequence of modules for raw image averaging](#)
 - [Sequence of modules for calibrated image or ndarray averaging](#)
- [Common mode correction in pnCCD ndarray](#)
- [Data corrections in module `ImgAlgos.NDArrCalib`](#)
 - [List of parameters in configuration file](#)
 - [2014-02-10 Test of the module `ImgAlgos.NDArrCalib`](#)
 - [Raw data](#)
 - [Pedestals](#)
 - [Common mode](#)
 - [Pixel status](#)
 - [Mask](#)
 - [Background](#)
 - [Gain](#)
 - [N*RMS Threshold](#)
 - [Common Threshold](#)
- [Test for amob33314 run 167](#)
 - [Evaluate parameters in Calibration Manager](#)
 - [Raw data average and rms](#)
 - [Pedestals subtracted average and rms](#)
 - [Pedestals and common mode subtracted average and rms](#)
- [Support of pnCCD in Calibration Manager](#)
- [Get latest version of calibman](#)
- [Image of pnCCD](#)
- [References](#)

pnCCD overview

[Large area pnCCD DAQ and Electronics](#), Lothar Struder & Robert Hartmann

Data for tests

On 2014-01-27 Sebastian Carron kindly provide us with data files for pnCCD experiment amoa1214:

- Dark Run: 169, rear sensors gain 1/64, front 1/1, Imaging mode exp=amoa1214:run=169
- Run With Hits: 170 Low hit rate though, so you will have to use a hit finder of sorts exp=amoa1214:run=170

Psana modules for pnCCD

Module `ImgAlgos.PncddNDArrProducer`

- Get from the event store `Psana::PNCCD::FramesV1`,
- Put in the event store `ndarray<const T,3>`, where `shape=[4][512][512]`, `T=uint16_t, int, float, double, int16_t`

Performance: ~13 ms/event

Modified module `ImgAlgos.PncddImageProducer`

- Get from the event store `Psana::PNCCD::FullFrameV1` or `ndarray<const T,3>` or `ndarray<const T,3>` for source and key parameters
- Put in the event store `ndarray<const T,2>`, where `shape=[1024+gap][1024]`, `T= input type`

Performance: ~30 ms/event (copy involves inverse iteration for 180 degree rotation of two bottom frames)

Sequence of modules for raw image averaging

- `ImgAlgos.PnccdImageProducer` - get `Psana::PNCCD::FullFrameV1`, put `ndarray<const uint16_t, 2>`
- `ImgAlgos.NDArrAverage` - averages `ndarray<const T, 2>`, save in file

Sequence of modules for calibrated image or ndarray averaging

- `ImgAlgos.PnccdNDArrProducer` - get `Psana::PNCCD::FramesV1`, put `ndarray<const T, 3>`
- `ImgAlgos.NDArrCalib` - getinput (raw) `ndarray<const T, Ndim>`, put `calibrated ndarray<const T, Ndim>`
- `ImgAlgos.PnccdImageProducer` - get `ndarray<const T, 3>`, put `ndarray<const T, 2>`
- `ImgAlgos.NDArrAverage` - averages `ndarray<const T, 2>` or `ndarray<const T, 3>`, save in file

Common mode correction in pnCCD ndarray

pnCCD image has intensity "strips" in both dimensions;

`[4][512][512]` array for single event and averaged over 1000 events:

At large number of events common mode should be averaged out. For 1000 events horizontal intensity "stripes" have gone.

This proves that common mode should be evaluated for horizontal stripes.

Due to specifics of pnCCD DAQ readout, it is recommended to evaluate common mode for consecutive groups of 128 pixel in the `[4][512][512]` array.

Data corrections in module `ImgAlgos.NDArrCalib`

Module description: [Module `ImgAlgos::NDArrCalib`](#)

List of parameters in configuration file

Module [`ImgAlgos::NDArrCalib`](#) is controlled by the list of parameters in the configuration file for psana, for example:

```
[ImgAlgos.NDArrCalib]
source = DetInfo(Camp.0:pnCCD.0)
key_in = pnccd-ndarr
key_out = calibrated
do_peds = yes
do_cmod = no
do_stat = no
do_mask = no
do_bkgd = no
do_gain = no
do_nrms = no
do_thre = yes
fname_bkgd =
fname_mask =
masked_value = 0
threshold_nrms = 0
threshold = 100.0
below_thre_value = 50
bkgd_ind_min = 0
bkgd_ind_max = 1000
bkgd_ind_inc = 2
print_bits = 11
```

2014-02-10 Test of the module `ImgAlgos.NDArrCalib`

Dark run: `exp=amoa1214:run=7`, all plots are shown for `Camp.0:pnCCD.0 event 5`

pedestals and pixel_rms are generated by the calibman for arrays of shape=[4,512,512] using the same run.

Raw data

All `do_* = no` - that means no correction is applied

Pedestals

```
do_peds = yes
```

file with pedestals is loaded automatically from

```
/reg/d/psdm/AMO/amoal214/calib/PNCCD::CalibV1/Camp.0:pnCCD.0/pedestals/1-end.data
```

Common mode

```
do_peds = yes
```

```
do_cmod = yes
```

files with pedestals and common_mode are loaded automatically from

```
/reg/d/psdm/AMO/amoal214/calib/PNCCD::CalibV1/Camp.0:pnCCD.0/pedestals/1-end.data
```

```
/reg/d/psdm/AMO/amoal214/calib/PNCCD::CalibV1/Camp.0:pnCCD.0/common_mode/1-end.data
```

where common mode parameters were set preliminary as:

```
echo "1 300 50 256 0.2" > /reg/d/psdm/AMO/amoal214/calib/PNCCD::CalibV1/Camp.0:pnCCD.0/common_mode/1-end.data
```

Average over each consecutive group of 256 pixels

Common mode subtraction improves the width of intensity distribution.

Pixel status

```
do_stat = yes
```

```
masked_value=0
```

Calibration type: **pixel_status** (0-good, 1,2,4,...-bad)

File with pixel status mask was produced in [Calibration Manager ROI Mask](#) application

```
/reg/d/psdm/AMO/amoal214/calib/PNCCD::CalibV1/Camp.0:pnCCD.0/pixel_status/7-7.data
```

Set bad pixels (1) in the half of frame[1]:

Mask

Region of interest (ROI) mask can be generated by the [Mask Editor](#).

```
do_mask = yes
```

```
fname_mask = pnccd-test-mask.txt
```

```
masked_value=0
```

Background

```
do_bkgd = yes
```

```
fname_bkgd = pnccd-test-mask.txt
```

For this test the file with pedestals is used:

Gain

```
do_gain = yes
```

file with gain factors is loaded automatically from

```
/reg/d/psdm/AMO/amoal214/calib/PNCCD::CalibV1/Camp.0:pnCCD.0/pixel_gain/7-7.data
```

For this test all gains for [4,512,512] pixels were set to 0.5

N*RMS Threshold

```
do_nrms = yes
```

```
below_thre_value = 0
```

```
threshold_nrms = 0.5
```

file with rms values is loaded automatically from

```
/reg/d/psdm/AMO/amoal214/calib/PNCCD::CalibV1/Camp.0:pnCCD.0/pixel_rms/1-end.data
```

Common Threshold

```
do_thre = yes
```

```
below_thre_value = 0
```

```
threshold = 100
```

Test for amob33314 run 167

Evaluate parameters in Calibration Manager

for types

- pedestals
- pixel_status for rms threshold = 15 ADU
- pixel_rms

Camp.0.pnCCD.0

Camp.0.pnCCD.1

Raw data average and rms

Pedestals subtracted average and rms

Pedestals and common mode subtracted average and rms

```
echo "1 50 10 128 0.2" > /reg/d/psdm/AMO/amob3313/calib/PNCCD::CalibV1/Camp.0:pnCCD.0/common_mode/167-167.data
```

Support of pnCCD in Calibration Manager

In dark run processing in [Calibration Manager](#) produces `pedestals` and `pixel_rms`. Then, if thresholds on rms and averaged intensity are set correctly, the `pixel_status` can be also produced and deployed under the `calib` directory. Calibration manager works with arrays of shape=[4,512,512]. Embedded ROI Mask Editor can be used to generate the ROI mask for pnCCD.

Get latest version of calibman

Run Calibration Manager from current release (for release version `ana-0.10.12`):

```
ssh -Y psana
cd <your-favorite-NON-RELEASE-directory>
sit_setup
calibman
```

If CalibManager or other packages were recently updated and these updates are wanted to be used:

```
ssh -Y psana
cd <your-favorite-directory>
newrel ana-current <release-directory>
cd <release-directory>
sit_setup

addpkg CalibManager HEAD;
addpkg <package-name-2> HEAD;
addpkg <package-name-3> HEAD;
...
scons;

calibman
```

Image of pnCCD

pnCCD image can be reconstructed from data by two methods using

1. [Module ImgAlgos::PnccdImageProducer](#) and
2. [Module ImgAlgos::NDArrImageProducer](#)

Examples of the first method are available in [psana - Module Examples](#)

Second method is available since release `ana-0.13.14` (package `PSCalib V00-02-40`).

It allows to reconstruct image using generic [Detector Geometry](#) technique with `geometry` file. Current version of pnCCD has a solid structure without moving parts, hence the main part of the geometry file is unchanged. Example of the geometry file for pnCCD can be found in `/reg/g/psdm/detector/alignment/pncdd/amo-pncdd.1-2015-01-23/calib/PNCCD::CalibV1/Camp.0:pnCCD.1/geometry/0-end.data`, which essential part is

#	HDR	PARENT	IND	OBJECT	IND	X0[um]	Y0[um]	Z0[um]	ROT-Z	ROT-Y	ROT-X	TILT-Z	TILT-Y	TILT-X
DETPNCCD:V1		0		PNCCD:V1	0	0	0	0	0	0	0	0.0	0.0	0.0
DETPNCCD:V1		0		PNCCD:V1	1	76725	38325	0	180	0	0	0.0	0.0	0.0
DETPNCCD:V1		0		PNCCD:V1	2	76725	77925	0	180	0	0	0.0	0.0	0.0
DETPNCCD:V1		0		PNCCD:V1	3	0	39600	0	0	0	0	0.0	0.0	0.0
#IP:V1		0		DETPNCCD:V1	0	38400	39000	10000	-90	0	0	0.0	0.0	0.0

The last commented line in this file can be un-commented in order to rotate and translate detector relative to IP.



- The name of the segment PNCCD:V1 should be unchanged. Other names DETPNCCD:V1 and IP:V1 are optional.
- Four segment coordinates are given in μm and it is assumed that each segment has 512×512 pixels of size $75 \times 75 \mu\text{m}^2$.
- All coordinates are given in matrix-like frame, where X axis goes from top to bottom, Y - from left to right.
- Gap is set to 16 pixels and may be a subject of calibration.

Example of the configuration file `psana-amoa1214-r0108-pnccd-NDarrImageProducer.cfg`

```
[psana]
# psana -m EventKeys -n 3 exp=amoa1214:run=108
#calib-dir = /reg/d/psdm/AMO/amoal214/calib
files = exp=amoa1214:run=108
events = 10

modules = ImgAlgos.PnccdNDarrProducer \
          ImgAlgos.NDarrCalib \
          ImgAlgos.NDarrImageProducer \
          ImgAlgos.NDarrAverage:nda \
          ImgAlgos.NDarrAverage:img-geo \
          ImgAlgos.Tahometer

[ImgAlgos.PnccdNDarrProducer]
source = DetInfo(Camp.0:pnCCD.1)
key_in =
key_out = pnccd-ndarr
outtype = asdata
print_bits = 0

[ImgAlgos.NDarrCalib]
source = DetInfo(Camp.0:pnCCD.1)
key_in = pnccd-ndarr
key_out = calibrated
do_peds = yes
do_cmod = yes
do_stat = yes
do_mask = no
do_bkgd = no
do_gain = no
do_nrms = no
do_thre = no
#fname_mask = pnccd-test-mask.txt
#fname_bkgd = pnccd-test-bkgd.txt
masked_value = 0
threshold_nrms = 4.0
threshold = 100
below_thre_value = 0
bkgd_ind_min = 10000
bkgd_ind_max = 10200
bkgd_ind_inc = 1
print_bits = 1

[ImgAlgos.NDarrAverage:nda]
source = DetInfo(Camp.0:pnCCD.1)
```

```

key          = calibrated
avefile      = pnccd-nda-ave
rmsfile      = pnccd-nda-rms
#maskfile    = pnccd-nda-msk
#hotpixfile  = pnccd-nda-hot
thr_rms_ADU  = 0
thr_min_ADU  = 2
thr_max_ADU  = 65000
print_bits   = 29

[ImgAlgos.NDArrImageProducer]
#calibdir = ./calib
calibdir = /reg/g/psdm/detector/alignment/pnccd/amo-pnccd.1-2015-01-23/calib
source     = DetInfo(Camp.0:pnCCD.1)
key_in     = calibrated
key_out    = pnccd-img-geo
print_bits = 1

[ImgAlgos.NDArrAverage:img-geo]
source     = DetInfo(Camp.0:pnCCD.1)
key        = pnccd-img-geo
avefile    = pnccd-img-geo-ave
rmsfile    = pnccd-img-geo-rms
#maskfile  = pnccd-img-geo-msk
#hotpixfile = pnccd-img-geo-hot
thr_rms_ADU = 0
thr_min_ADU = 2
thr_max_ADU = 65000
print_bits = 29

[ImgAlgos.Tahometer]
dn         = 100
print_bits = 7

```

uses psana modules

- `ImgAlgos.PnccdNDArrProducer` - gets pnCCD data and put it as ndarray in the event store
- `ImgAlgos.NDArrCalib` - applies calibrations to ndarray
- `ImgAlgos.NDArrImageProducer` - produces calibrated image
- `ImgAlgos.NDArrAverage:nda` - averages calibrated ndarray
- `ImgAlgos.NDArrAverage:img-geo` - averages calibrated image

This script can be executed by the command

```
psana -c psana-amoa1214-r0108-pnccd-NDArrImageProducer.cfg
```

which saves averaged and RMS files for calibrated ndarray and image, which can be plotted by the command `plims <file-name>`. For example

```
plims pnccd-nda-ave-amoa1214-r0108.dat
plims pnccd-img-geo-ave-amoa1214-r0108.dat
```

plot images with intensity distributions for ndarray and image, respectively:

Masks and other calibration files for pnCCD and image can be generated with `calibman`.

References

- [psana - Module Examples](#)
- [Psana Module Catalog - Old](#)
- [Calibration Management Tool](#)
- [Mask Editor](#)
- [CSPAD Calibration](#)
- [Large area pnCCD DAQ and Electronics, Lothar Struder & Robert Hartmann](#)

- [LAMP PNCCD User Manual](#)