EPIX10KA2M and EPIX10KAQUAD

Two new detectors EPIX10KA2M and EPIX10KAQUAD are composed from EPIX10KA modules.

Content

Geometry

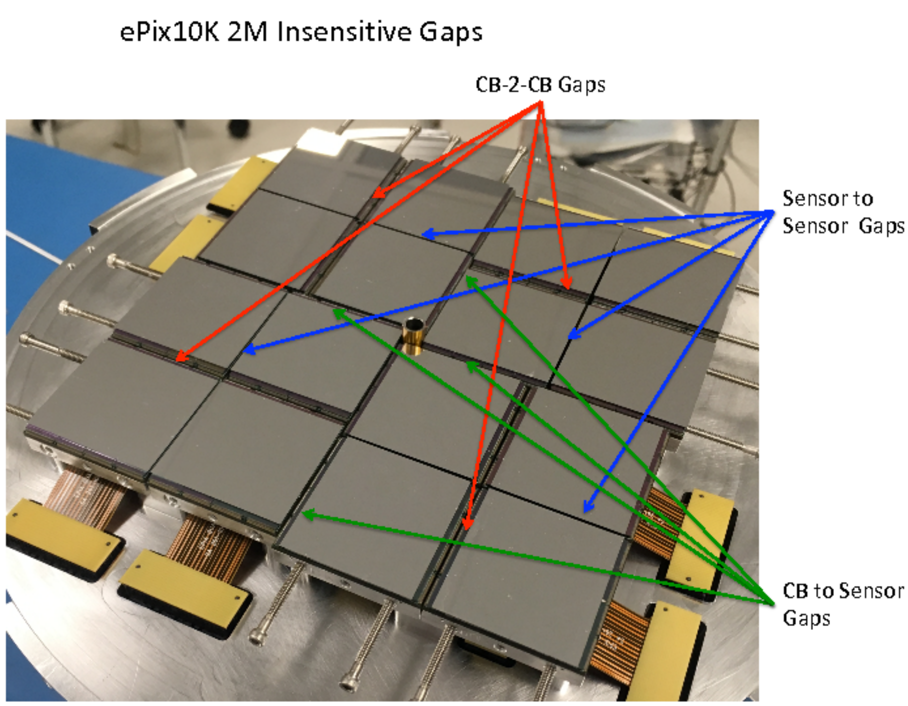
Plots and comments from Chris Kenny

Comments on epix10ka2m geometry from Chris Kenney

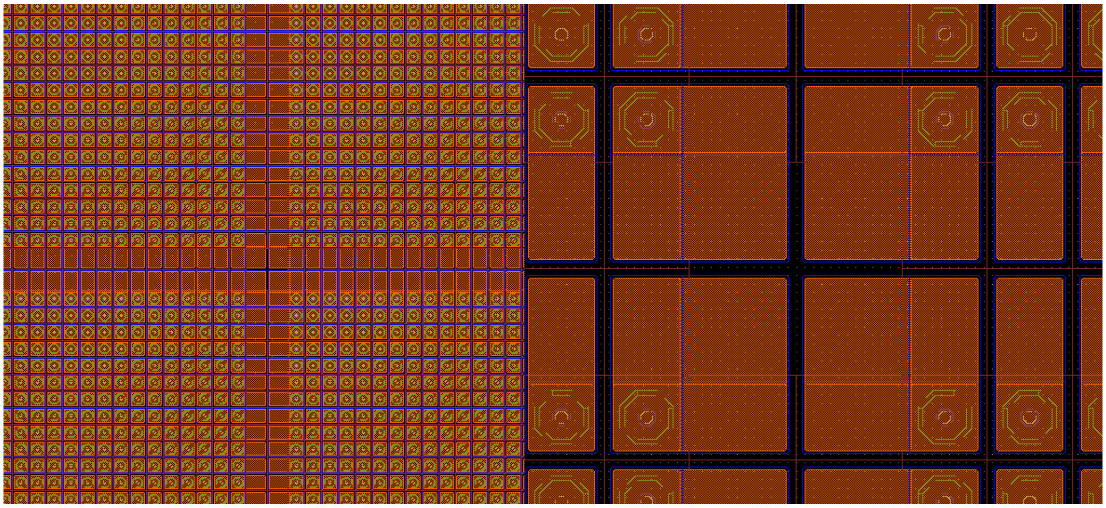
```
Kenney, Christopher J. 2018-11-12, 2:26 PM

Guide tube is 7mm outer diameter
We added polyimide tape that was 150 microns thick before application
to the tube.

But a decent estimate would be the mechanical edge-to-edge orthogonal
separation between sensor edges 7.3 mm.
We need to add about 1 mm for the guard rings on each sensor

So the orthogonal gap between active pixels on opposing quads across
the beam guide tube should be 8.3 mm

====
Kenney, Christopher J. 2018-11-12, 4:41 PM Blaj, Gabriel;Dubrovin, Mikhail;Kwiatkowski, Maciej

Very rough estimate of the gaps
There are 3 types of gaps
All are active pixel to active pixel
sensor to sensor ~ 1.6 mm
CB to CB ~ 6.4 mm
sensor to CB ~ 3.9 mm

====
CB = Carrier Board edges
Full camera image below
====
so 384 columns parallel to the balcony (the widest dead gaps)
and 352 orthogonal to the balcony (vertical direction)
```

epix10ka2m assembly

epix10ka2m-insensitive-gaps.pdf



epix10ka sensor central region between 4 ASICs

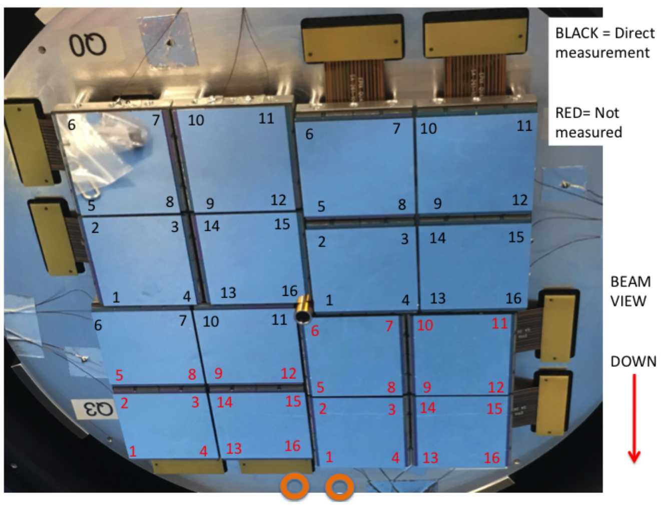


The internal gap between four ASICs in sensor,

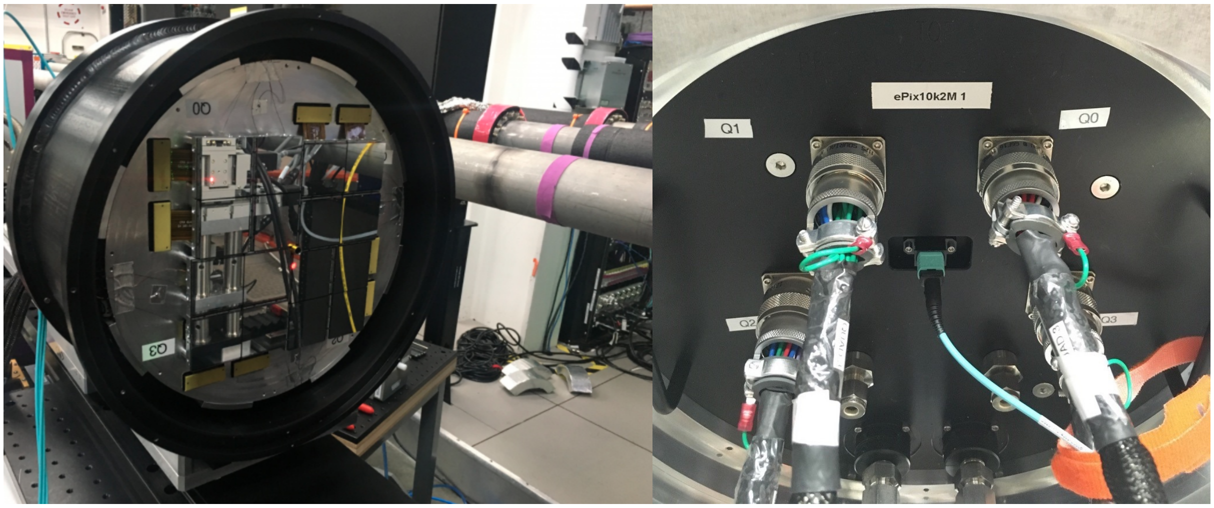
pixel size 100 x 225 microns in area in both directions.

Metrology map from Chris Kenney

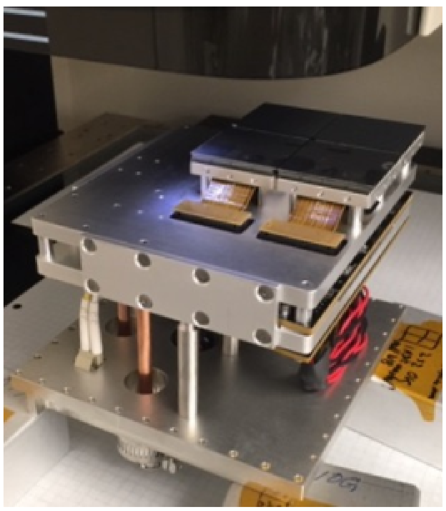
- 2018-11-18-epix10ka2m-metrology-map.pdf
- 2018-11-18-epix10ka2m-metrology-map.pptx
- 2018-11-15-Metrology-epix10ka2m.xlsx



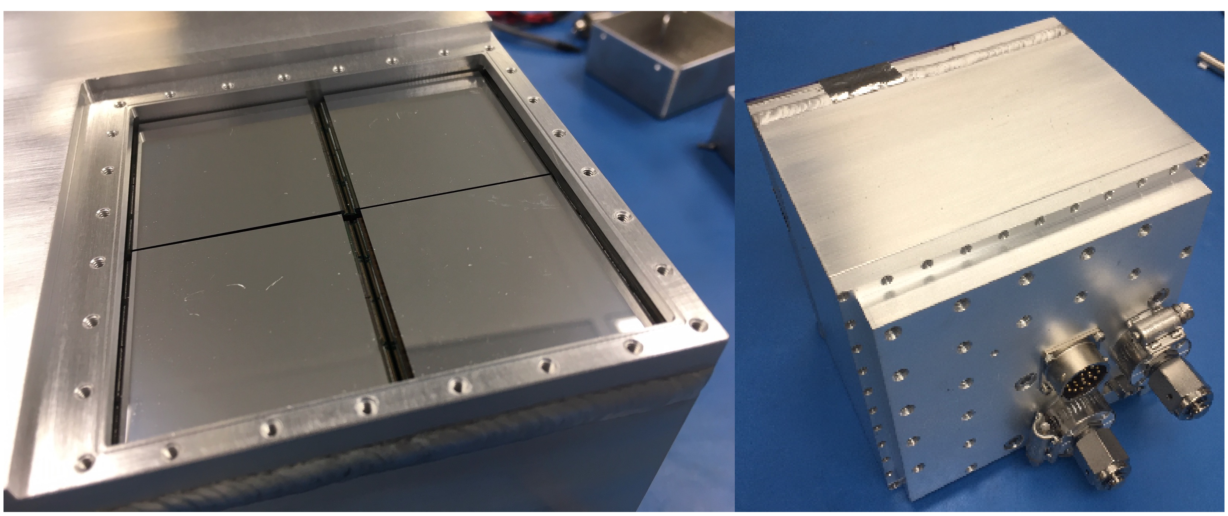
Front and back side of the new detector epix10ka2m.1 from mfxc00118 2020-07-dd



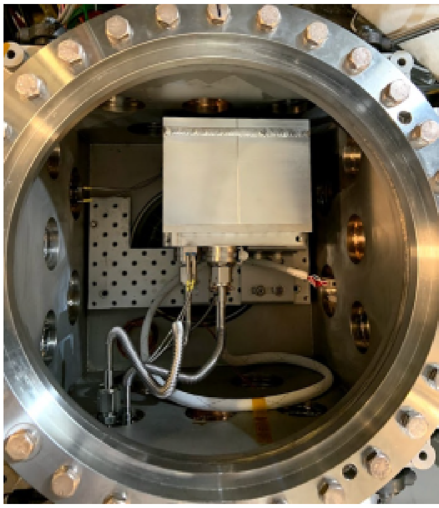
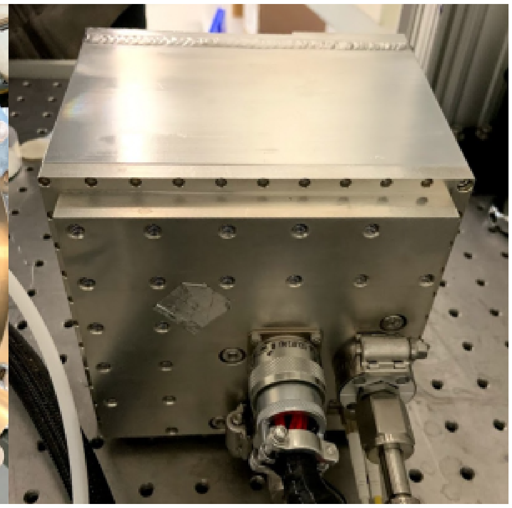
2020-10-04 MEC epix10kaquad 0 at optical metrology



2020-10-06 MEC epix10kaquad 0 front and back



2021-04-10 UED epix10kaquad



Comments on orientation of epix10ka2m parts from Matt

2018-11-19 orientation schema

```
Weaver, Matt
2018-11-19, 7:37 PMO'Grady, Paul Christopher;Dubrovin, Mikhail

Hi Mikhail,
We took radioactive source test
 runs today to verify the geometry.  We found that we were rotated by 90
 degrees, which matches the labeling on the back of the detector as well
 as the metrology picture from Chris Kenney.
The runs are
/reg/d/psdm/det/detdaq17/e968-r0131   - source unmasked (up to ~ event 10000), then masked vertically after ~
event 15000
/reg/d/psdm/det/detdaq17/e968-r0132   - source masked horizontally at bottom after ~ event 15000.

So, the picture is...
  //  (Epix10ka2m)
  //         |
  //  Quad 0 | Quad 1      Quad 2 is rotated  90d clockwise
  //  -------+--------     Quad 3 is rotated 180d clockwise
  //  Quad 3 | Quad 2      Quad 0 is rotated 270d clockwise
  //         |
  //
  //  (Quad 1)
  //         |
  //  Elem 0 | Elem 1
  //  -------+--------     No rotations
  //  Elem 2 | Elem 3
  //         |
  //
  //  (Elem 0)
  //         |
  //  ASIC 0 | ASIC 3
  //  -------+--------     No rotations
  //  ASIC 1 | ASIC 2
  //         |
  //
  //  (Elem 0-3 pixel array)
  //                    row increasing
  //                          ^
  //                          |
  //                          |
  //  column increasing <-- (0,0)
```

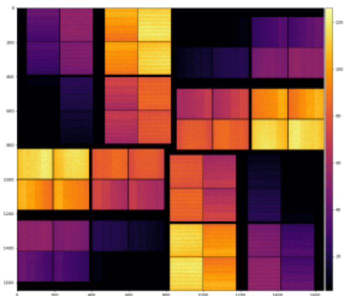
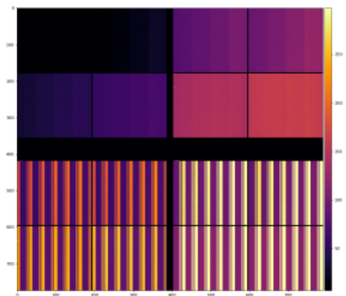
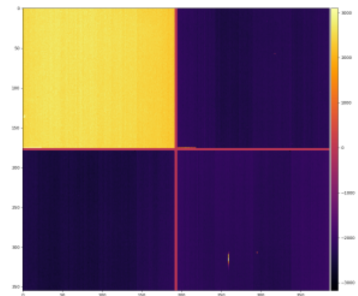
Preliminary geometry

in /reg/g/psdm/detector/data_test/calib/

```
/reg/g/psdm/detector/data_test/calib/Epix10ka2M::CalibV1/NoDetector.0:Epix10ka2M.0/geometry/0-end.data @
(epix10ka2m - entire detector)
/reg/g/psdm/detector/data_test/calib/Epix10kaQuad::CalibV1/NoDetector.0:Epix10kaQuad.0/geometry/0-end.data @
(epix10kaquad - one quad)
/reg/g/psdm/detector/data_test/calib/Epix10ka::CalibV1/MecTargetChamber.0:Epix10ka.1/geometry/0-end.data @
(epix10ka - one panel)
```

copy of geometry files in alignment examples /reg/g/psdm/detector/alignment/

```
/reg/g/psdm/detector/alignment/epix10ka2m/calib/Epix10ka2M::CalibV1/NoDetector.0:Epix10ka2M.0/geometry/0-end.
data
/reg/g/psdm/detector/alignment/epix10kaquad/calib/Epix10kaQuad::CalibV1/NoDetector.0:Epix10kaQuad.0/geometry/0-
end.data
/reg/g/psdm/detector/alignment/epix10ka/calib/Epix10ka::CalibV1/MecTargetChamber.0:Epix10ka.1/geometry/0-end.
data
```

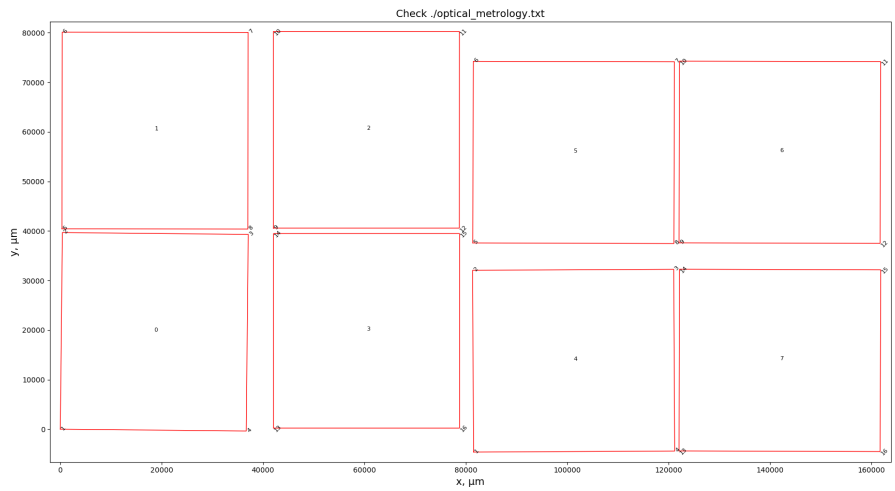
Optical metrology processing

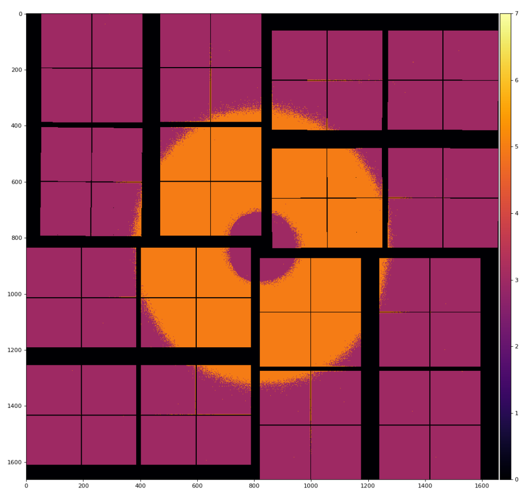
Scripts for processing

```
CalibManager/app/
  optical_metrology_check
  optical_metrology_epix10ka2m
```

Results in

```
/reg/g/psdm/detector/alignment/epix10ka2m/calib-mfx-epix10ka2m-01-2018-11-15/
  2018-11-15-Metrology-epix10ka2m.xlsx
  2018-11-15-Metrology-epix10ka2m.txt
  2018-11-15-Metrology-epix10ka2m-corr.txt
  2018-11-15-geometry-epix10ka2m.txt - geometry file accounting for optical metrology data
  README-2018-11-15
```





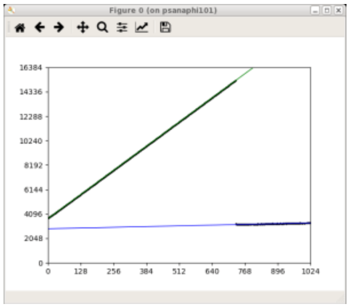
Gain

Gabriel's comments on gain factors

```
Blaj, Gabriel
2018-12-04, 2:08 PMO'Grady, Paul Christopher;Nelson, Silke;Dubrovin, Mikhail;Hart, Philip Adam

Hi,
You could try to use the gain files obtained with the pulser. They are not great but might work.

For a better gain calibration, we should use single photon data.
There is sufficient 1 photon data taken during the first testing at XCS,
but it will take me a few days to calculate the gains.

I would actually advocate returning the number of photons (as we
discussed in a meeting a few months ago). Even without a calibration it
can be easily calculated from the (average) gains:

High (FH and AHL): 132 ADU/9.5 keV
Medium (FM and AML): 43 ADU/9.5 keV
Low (FL, AHL, AML): 1.32 ADU/9.5 keV

(Just a note, while the pulser is not great for calibrating gains, it works fine for offset calibration)

Thanks,
Gabriel
```

Gain factors from charge injection default and measured

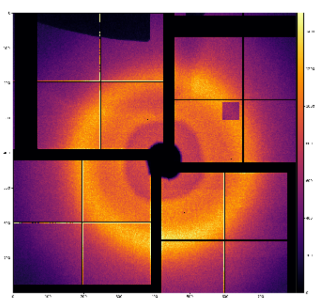
gain	charge injection	current default	measured (ADU / keV)	2020-08-03 Gabriel (ADU / keV) - use as default
L	0.46	0.01	0.139	0.164
M	15.	0.3(3)	4.5	5.466
H	46.7	1	13.9	16.40

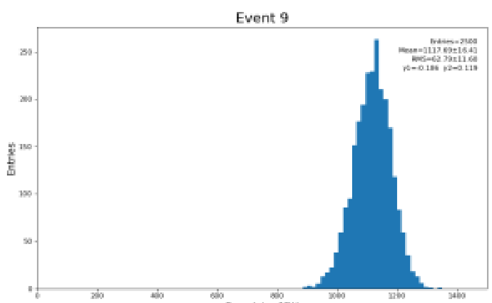


Gain factors default vs charge injection

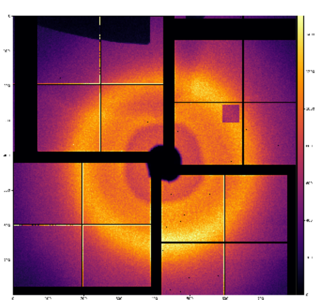
- Detector/examples/ex_epix10ka_images.py
- XcsEndstation.0:Epix10ka2M.0
- charge injection gain factors were generated from exp=xcsx35617:run=544
- data with water ring for comparison exp=xcsx35617:run=528
- account relative factor 46.7
- selected rect [6, 120:170, 200:250]

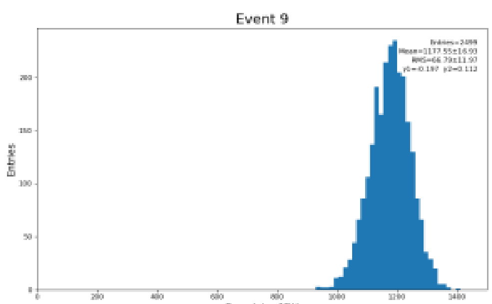
gain default: H / M / L = 1 / 0.33333 / 0.01





gain from charge injection:





constants	Mean	RMS	RMS / MEAN
default	1117.7	62.72	0.05618
charge injection	1177.5	66.79	0.05672

Conclusion: in this test charge injection **gainci** constants do not improve gain factors comparing to default

Default gain correction factors

2020-08-03 Gabriel Blaj about ADU/keV default gain factors

```
Blaj, Gabriel <blaj@slac.stanford.edu> Mon 8/3/2020 6:52 PM
To: Hart, Philip Adam;
 Dragone, Angelo;
 Kenney, Christopher J.;
 Dubrovin, Mikhail;
 O'Grady, Paul Christopher;
 Hansson, Conny;
 McKelvey, Mark E

Hi, Here are some good starting values for the ADC to keV conversion:

High gain: 132 ADU / 8.05 keV = 16.40 ADU/keV
Medium gain: 132 ADU / 8.05 keV / 3 = 5.466 ADU/keV
Low gain: 132 ADU / 8.05 keV / 100 = 0.164 ADU/keV

Of course, a gain calibration is preferable.
The same numbers work in both fixed and auto-ranging gain modes.
Thanks,
Gabriel

=========
Blaj, Gabriel <blaj@slac.stanford.edu> Mon 8/3/2020 7:13 PM

Hi, For the long integration time, I don't have a set of magic numbers, but this iterative procedure should
yield optimal settings:

Cool the camera as low as possible, just a few degrees over the minimum temperature to allow temperature
stabilization by the PID control loop (either the chiller PID for the large cameras, or the Peltier PID in the
small cameras). Of course, the small cameras can be cooled much lower than the large ones.


Start with the default LCLS settings (I believe both AsicAcqWidth and R0toAcq are set to 100us by default)

0 AsicAcqWidth should be optimized for the experiment. With a very cold camera (e.g., < -15ºC) you could go to
5ms. A good starting value would be 1ms.

1 Set AsicAcqWidth to, e.g., 1 ms
2 Set R0toACQ time to 100us
3 Decrease frame rate until no frames are dropped
4 Set the X-ray source to a low flux (0.01-0.05 photons/pixel/frame?)
5 Try to get a uniform illumination
6 Repeat:
- Calibrate dark
- Take many frames and integrate them
- Look if the resulting image is uniform or has a strange sawtooth pattern over each ASIC
- If no, try reducing R0toACQ
- If yes, try increasing R0toAACQ
- Increase/decrease frame rate to the maximum frame rate that runs reliably (no dropped frames).
6 Until an optimum is found.

For an idea how the strange sawtooth pattern looks, you could try setting:
AsicAcqWidth = 1ms
 R0toAcq = 50us, or 20us.

Thanks,
Gabriel
```

Test of the gain switching modes

offset calibration: exp=xcsx35617:run=544; its timestamp 20181129124822 faked for earlier dark calibrations by reference from 20180101000000

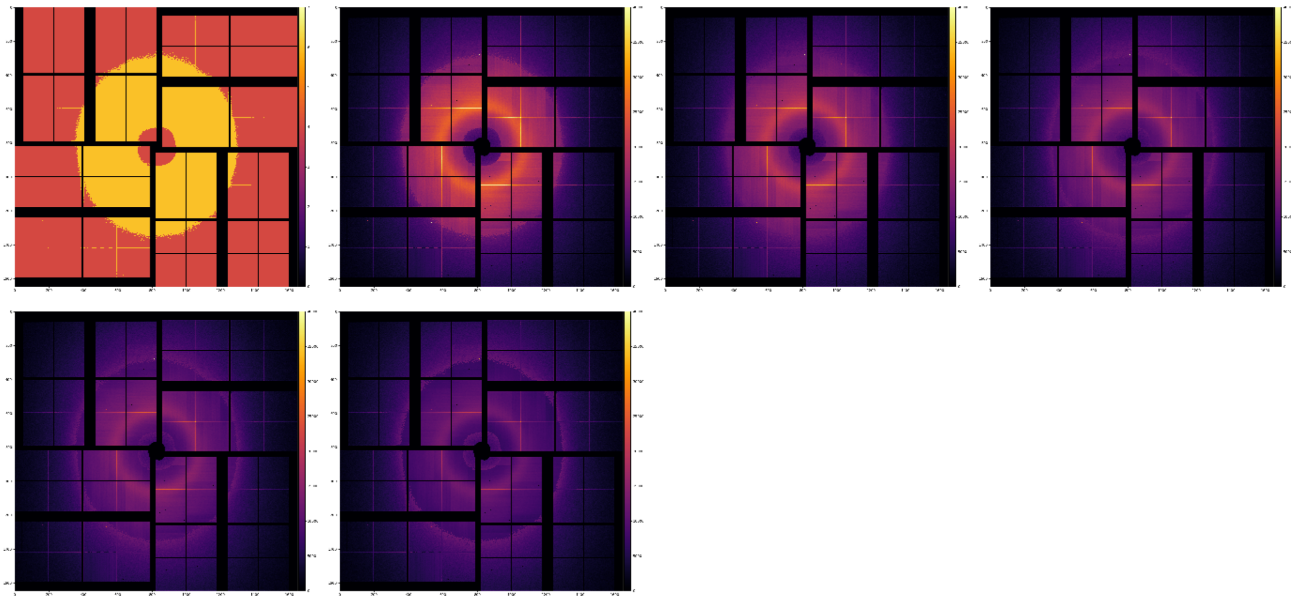
dark runs: 413, 416, 417, 420 of xcsx35617

gain factors M, H=1, L= 0.2, 0.25, 0.3, 0.33333, 0.4

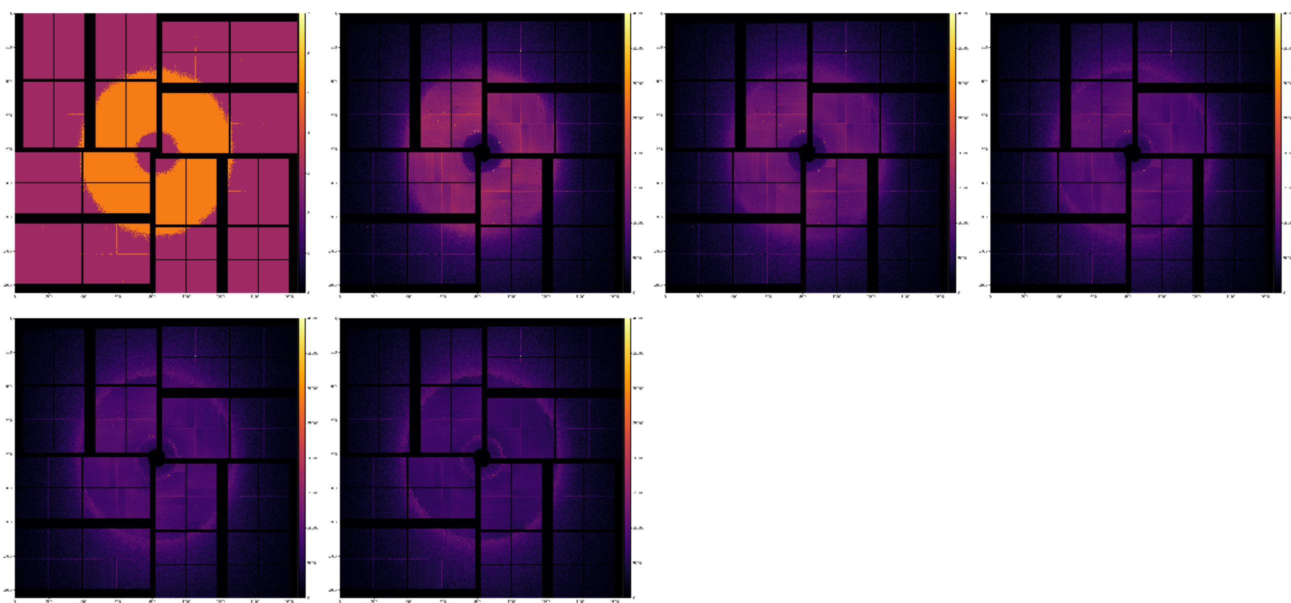
gain map images show that lateral and central-most pixels in mode H, M, "water ring" region pixels switched to L

data:

- AML: exp=xcsx35617:run=419, event 3



- AHL: exp=xcsx35617:run=414, event 3



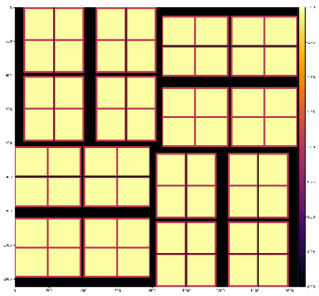
Masks

mask_geo

mask_geo = det.mask_geo(par, mbits=3, width=10, wcentral=5)

- mbits = 1 - masks edges, +2 - masks central rows and columns.
- width - number of edge rows or columns to mask, def=1
- wcentral - number of central rows or columns to mask, def=1

plot for mask_geo + 1:



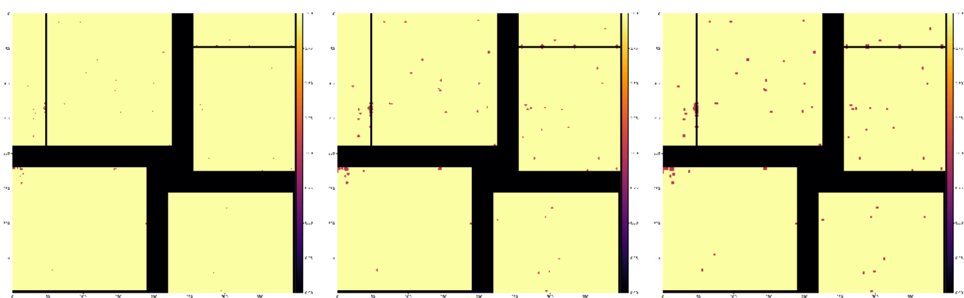
status_as_mask

- use pixel_status for exp=xcsx35617:run=544
- mask_status = det.status_as_mask(par, mode=0, indexes=(0,1,2,3,4))
- mode 0/1/2 masks zero/four/eight neighbors around each bad pixel
- indexes=(0,1,2,3,4) # indexes stand for FH, FM, FL, AHL-H, AML-M, respectively. Derived modes have the same status arrays.

found number of bad pixels

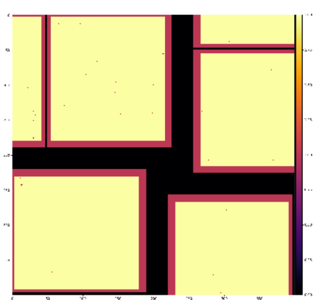
- 2802 for F gain modes and
- 3253 for all F + A mode

plots for mask_status + 1 for mode=0, 1 and 2:



Combined mask

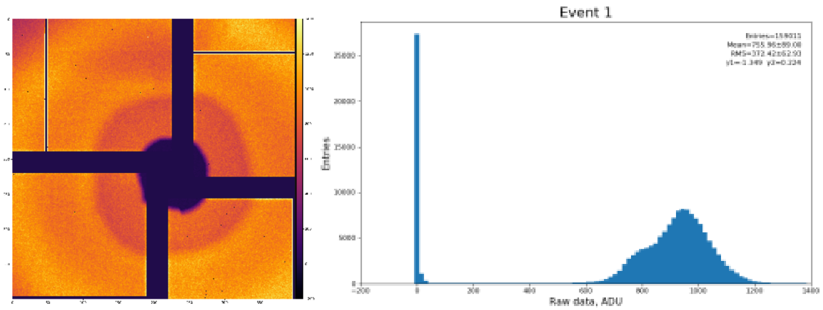
mask = det.mask(par, calib=False, status=True, edges=True, central=True, width=10, wcentral=5, mode=0)

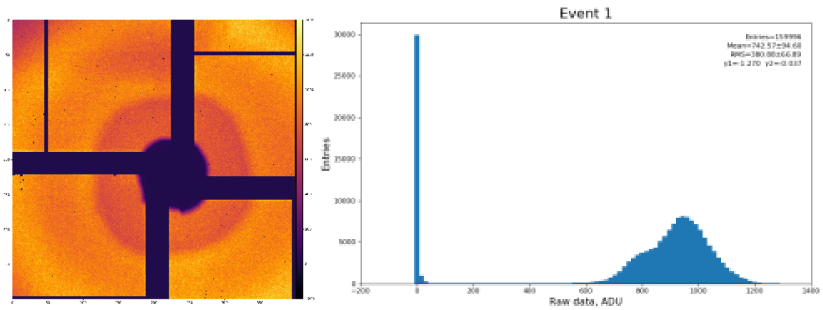


Calibrated data and mask

Image and spectrum for

- nda = calib_epix10ka_any(det, evt)
- nda *= det.mask(par, calib=False, status=True, edges=True, central=True, width=1, wcentral=1, mode=0)





Manual alignment on 2019-05-06

Data

Ring-data (npy) arrays were provided for xcsx35617 run 400 by Silke, available under

- /reg/g/psdm/detector/alignment/epix10ka2m/calib-mfx-epix10ka2m-01-2018-11-15/2019-05-06-geometry-alignment/

Alignment tool

Manual Detector alignment tool (geo) is used for alignment. There is no automated geometry optimization in this tool.

Initial geometry

Alignment is started with the best geometry file obtained after optical metrology measurements for two quads, like

/reg/g/psdm/detector/alignment/epix10ka2m/calib/Epix10ka2M::CalibV1/NoDetector.0:Epix10ka2M.0/geometry/geo-epix10ka2m-v180

or

/reg/d/psdm/xcs/xcsx35617/calib/Epix10ka2M::CalibV1/XcsEndstation.0:Epix10ka2M.0/geometry/398-398.data

Alignment procedure

Quads' x0,y0 - center positions ONLY have been tuned as explained here:

1) Q0 and Q1 were moved together relative to the image center, because their geometry is constrained from optical metrology.

2) then Q2 and Q3 were moved independently in order to get consistent "to my eye" image relative to a set of drown circles.

Geometry for panels inside Q2 and Q3 is set from design geometry, and I do not feel that could do better job moving panels in quad.

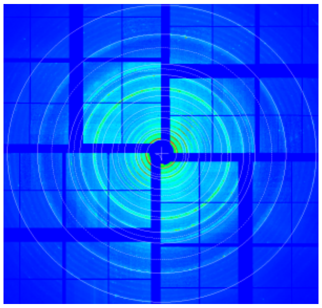
There are some regular alignment issues with this detector; if I tune nicely (with precision ~ pixel size) rings in the middle of radial range,

then internal and external rings may be misaligned. This may be due to small tilt of the detector or non-accounted z position of panels

w/o optical metrology.

Results

Resulting geometry for this data looks like on attached image.



All files are available under

- /reg/g/psdm/detector/alignment/epix10ka2m/calib-mfx-epix10ka2m-01-2018-11-15/2019-05-06-geometry-alignment/

Recommendation for further geometry improvement

The only reliable procedure to get correct detector geometry is an 3-d optical metrology of entire detector.

After that one would need to adjust precisely

1) detector center relative to image with rings

2) sample-to-detector distance

3) detector plane tilts.

References

- EPIX10KA2M References
- EPIX10KA
- EPIX10KA2M data images and geometry
- Production of calibration constants for multi-panel epix10ka detectors
- EPIX10KA2M Charge injection fit issue
- EPIX10KA2M issue with panel 6 in mfxc00318
- EPIX10KA2M issue with panel 12 in xcsc00118 2M.0
- Optical metrology of epix10ka2m.1 from 2020-02-25
- EPIX10KA panel flatness measurement
- Jungfrau and Epix10ka Calibration
- Detector alignment tool
- 2020-08-20-new-x-ray-detector-snaps-1000-atomic-level-pictures-second-natures-ultrafast