




IDPE for TCE in XCS - Problems and tentative solutions




- Problems and tentative solutions
 - Pedestals
 - Low level threshold
 - Image filtering
 - Selection of intensity monitors
 - Dynamic mask
 - Static mask
 - Graphical editor for selected regions
 - Center of the image
 - Correct normalization of g2
 - GUI

Problems and tentative solutions

Based on 2012-10-03 meeting:

In order to be useful this application should do correct math, accounts for image mask, discards bad events (noisy and "bright" pixels), apply normalization etc., and have a convenient GUI. Below is a list of requirements, marked as , with suggested solutions, marked as  if exists or as  if needs to be implemented.

Pedestals

-  "dark" run number should be provided by user and the imaging camera pedestals should be evaluated and applied for all runs until the "dark" run number has not changed.
-  For pedestals evaluation: use available `ImgAlgos::ImgAverage` psana module for "dark" run, which produces file with averaged over events pedestals and the file with their rms values.
-  For pedestals subtraction: use `ImgAlgos::ImgCalib` psana module in the same job which evaluates correlators; the pedestals will be subtracted and corrected image will be retained in the event and used for correlator calculations.

Low level threshold








-  Image pixel intensity physically can't be negative. Low amplitude noise should be suppressed by the threshold. The threshold amplitude should be provided by user (along with substituting amplitude).
-  Is implemented in the `ImgAlgos::ImgCalib` psana module, right after pedestal subtraction.



Image filtering

-  Usually users use different type of intensity monitor signals in order to retain/discard image for/from further processing. Discarded images should not contribute into the correlation analysis. The spectra of intensity monitors should be available for browsing. User should be able to select the intensity monitor(s) from the list and set low and high thresholds.
-  The filtering module may be implemented in psana. Based on selected intensity monitor(s) and thresholds it will decide to retain or discard event and accumulate spectral histograms. The histograms will be saved in file at the end of run.
-  Control GUI should be able to browse the intensity monitor histograms and set the thresholds.

Selection of intensity monitors

-  It would be nice to have an algorithm like in XTC explorer
-  Possible options:
 - run application as a plug-in for XTC Explorer,
 - pyana module performing similar to XTC Explorer algorithm,
 - stand-alone C++ module reading XTC datagrams,
 - hardwired list of intensity monitors.
 - psana module, which needs to be developed

Dynamic mask

-  Imaging camera may have permanently hot pixels or some pixels may be saturated during the run. User need to set a thresholds on hot pixels and high intensity.
- If the pixel amplitude crosses the high intensity threshold at least once during the run, then this pixel should be excluded from further analysis. The same is valid for hot pixels, which shows above threshold intensity in large fraction of events.
-  This is implemented in the psana module `ImgAlgos::ImgMaskEvaluation`, which works before event selection algorithm. Two files of image size may be produced 1) for saturated and 2) for hot pixels.

Static mask

- ? The beam-stopper region and some areas with fringes should be masked. It could be useful to have a graphical editor for mask.
- + See section for [GUI](#).

Graphical editor for selected regions

- ? Sometimes it is useful to select good region of the image. It could be convenient to use a graphical editor, as for mask.
- + See section for [GUI](#).

Center of the image

- ? User should have an option to set a center of the rings for histograms.
- + See section for [GUI](#).

Correct normalization of g2

- ? Evaluation of g_2 for image **regions** is not that simple as presented by the formula for a single pixel:

$$g_2(\tau) = \langle I(t) * I(t+\tau) \rangle / (\langle I(t) \rangle * \langle I(t+\tau) \rangle),$$

In order to get physically meaningful results for g_2 , the correlators $\langle I(t) \rangle$ and $\langle I(t+\tau) \rangle$ should be averaged in the fine rings around center with number of bins N_2 , which is order of 100, with dR down to 1-2 pixels.

Then the $\langle I(t) * I(t+\tau) \rangle$? correlator should be averaged over bold rings intended for G_2 evaluation. The number of these rings N_1 should be order of 10. The N_2 and N_1 should be defined by user.

It might be useful to define the histogram region by the sector in the user-defined angular range.

+ In order to have required normalization of correlators, it is not enough to save the g_2 value only. So, the format of the resulting file has changed. Now for each value of τ the output file contains the $\langle I(t) \rangle$, $\langle I(t+\tau) \rangle$, and $\langle I(t) * I(t+\tau) \rangle$, each for entire image written in binary for float format. Not all masks, selection regions, normalization etc. are available during correlators calculation, so correlators are evaluated for all pixels. Which pixels should be included in the G_2 for each region can be decided at the final stage of processing. This approach allows to perform the most time consuming procedure - the correlators calculation once and do the analysis after that.

- + See section for [GUI](#).

GUI

? In order to get an easy interface to all sub-processes, it seems to be useful to have a GUI with configuration of everything through the GUI.

! Well, presumably users will want different specific features in their analyses which can not be foreseen in implementation of GUI. It is pretty unlikely that everything in analysis can be done clicking on buttons in GUI. Then, it could be nice if user understand what he is doing step by step and have a monitoring at the end of each stage. We are doing science, not a standard pre-defined things... Most generic way to process data is to have a separate procedures with command line interface.

+ Anyway, the browser/presenter of data stored in the files after pre-processing could be provided for a set of common plots.

All features listed in previous sections, such as static and dynamic mask, restriction of the region(s) of interest, selection of the image center, the binning scheme etc., can be done in the browser at the final stage of the analysis.

+ Generic GUI as a shell for entire analysis can be implemented as an interface to the command line procedures:

- Each command line procedure may have a dedicated GUI for procedure configuration and monitoring.
- All procedures may be listed in the main GUI with a status sign.
- The main issue of this approach is a cross-configuration between separate procedures. This can be achieved if all procedures will have a common list of configuration parameters.