Recognition of hot pixels on CCDs in the eROSITA Near Real Time Analysis

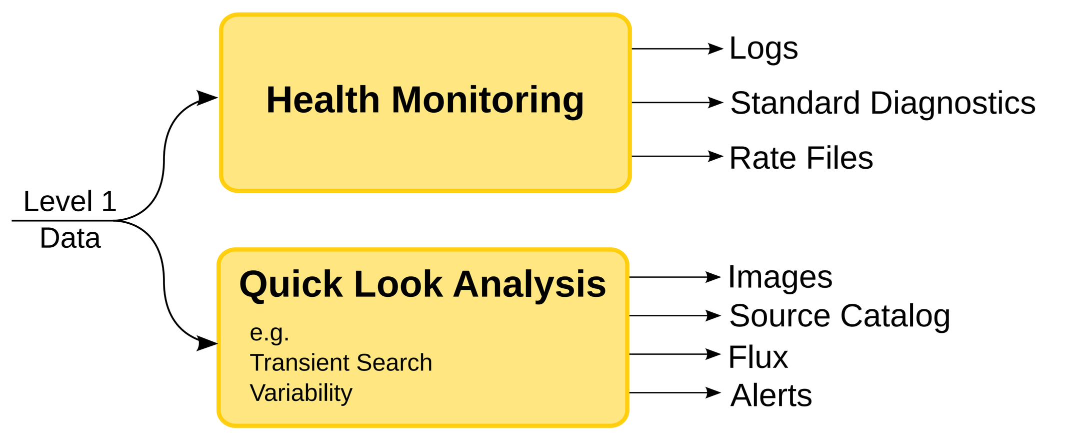
M. Wille[1], C. Großberger[1], C. Schmid[1], H. Brunner[2], J. Wilms[1], I. Kreykenbohm[1], J. Hölzl[1]
[1]Dr. Karl Remeis-Observatory & ECAP, Universität Erlangen-Nürnberg, Sternwartstraße 7, 96049 Bamberg, Germany
[2]Max-Planck-Institut für extraterrestrische Physik (MPE), Giessenbachstraße, 85748 Garching, Germany
eMail: michael.wille@sternwarte.uni-erlangen.de

Abstract

With the eROSITA Near Real Time Analysis (NRTA) we present a versatile software framework which easily permits sanity checks, first quick-look analysis and automated archiving of mission data. We give a short overview of the general working principle of the NRTA and show examples of a conceivable tool chain which can be used for data processing with special emphasis on the software responsible for the detection of defective pixels on charge-coupled devices (CCDs). We demonstrate the capabilities of the detection software (such as differentiation between bright pixels and bright lines as well as recognition of flickering pixels) and discuss the algorithm which is based on statistical analysis of raw event data. Afterwards we show the results of hot pixel search tests with data taken by XMM-Newton. The performance of the detection software suggests that it is capable of securely detecting hot pixels with a mean signal of $\sim 2.5\sigma$ above background level.

NRTA

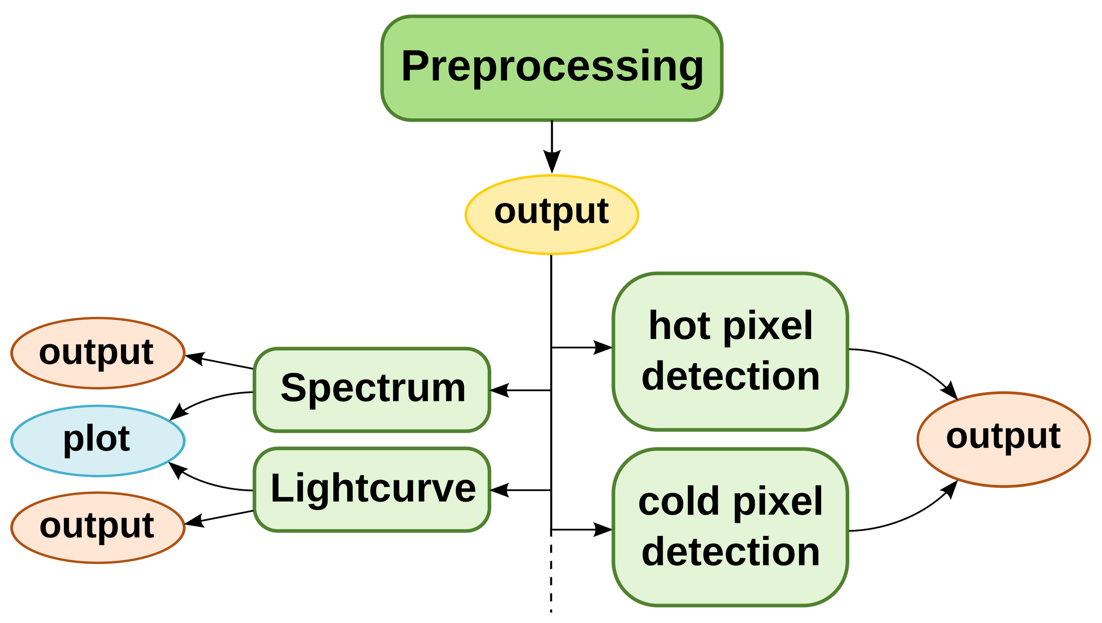
The primary purpose of the eROSITA NRTA (Near Real Time Analysis) Software is to monitor the health of the eROSITA instrument by performing checks on housekeeping parameters as well as generating summaries on telemetry data and searching for deviations from the nominal instrument status.



In addition, the NRTA generates quick look information of scientific value such as

- preliminary images of the sky observed during the current telemetry download
- search for transient phenomena in the data

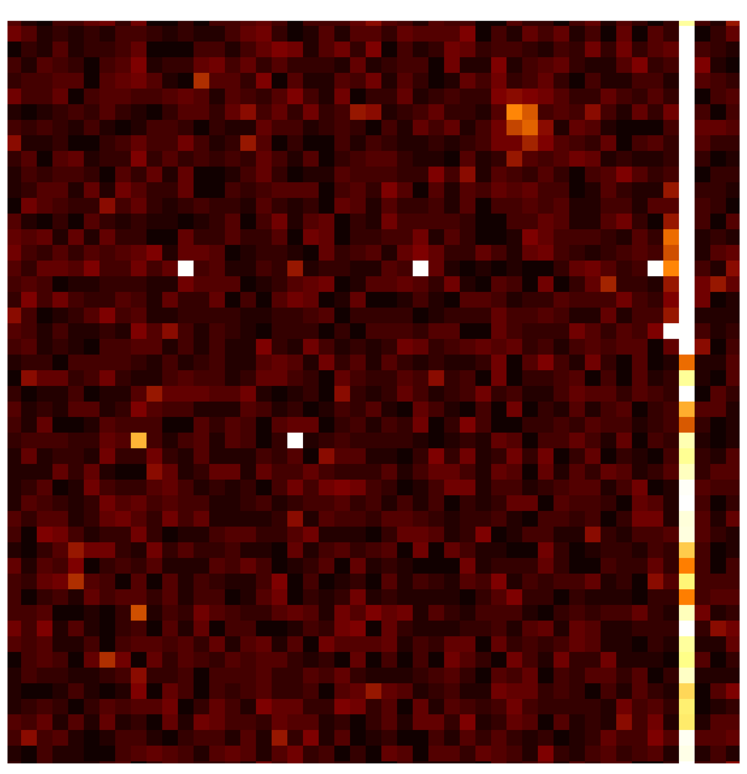
Due to its modular concept the NRTA is a versatile framework which can run many different tools in parallel and/or sequentially. Toolchains can easily be constructed via pipeline definitions in XML-format. Mutual dependencies and priorities between the tools are handled by semaphores. The figure below shows an example of a small pipeline performing a bad pixel detection and producing a lightcurve and a spectrum plot.



For further information about the NRTA Software see the poster of C. Großberger and I. Kreykenbohm.

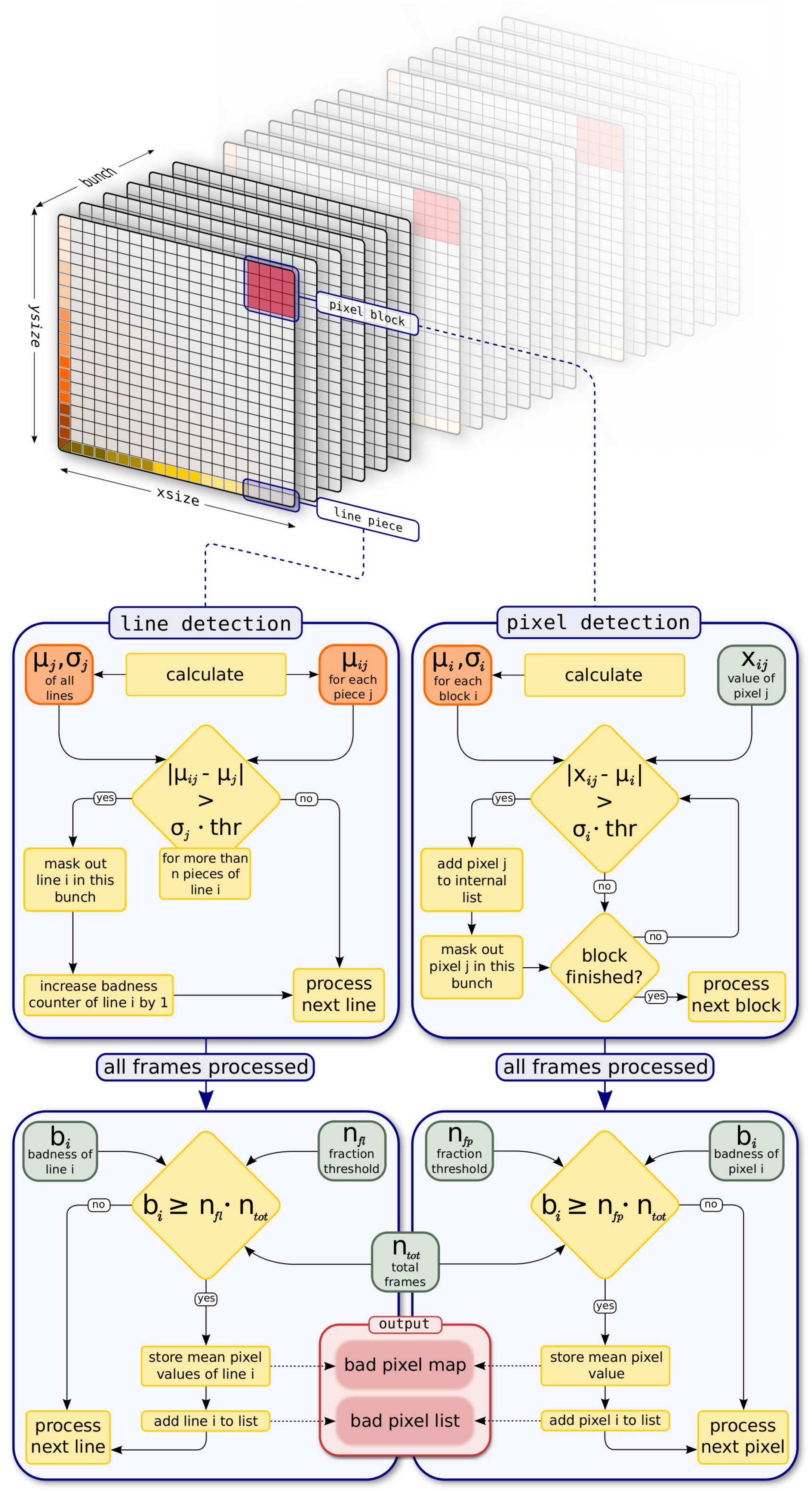
Hot pixels

Defects in the crystal structure of CCDs (such as, e.g., contamination with other elements) can lead to local changes in the electronical properties of the chip. This effect may cause pixels which show a signal much bigger (*hot pixels*) or less (*cold pixels*) than it should be.



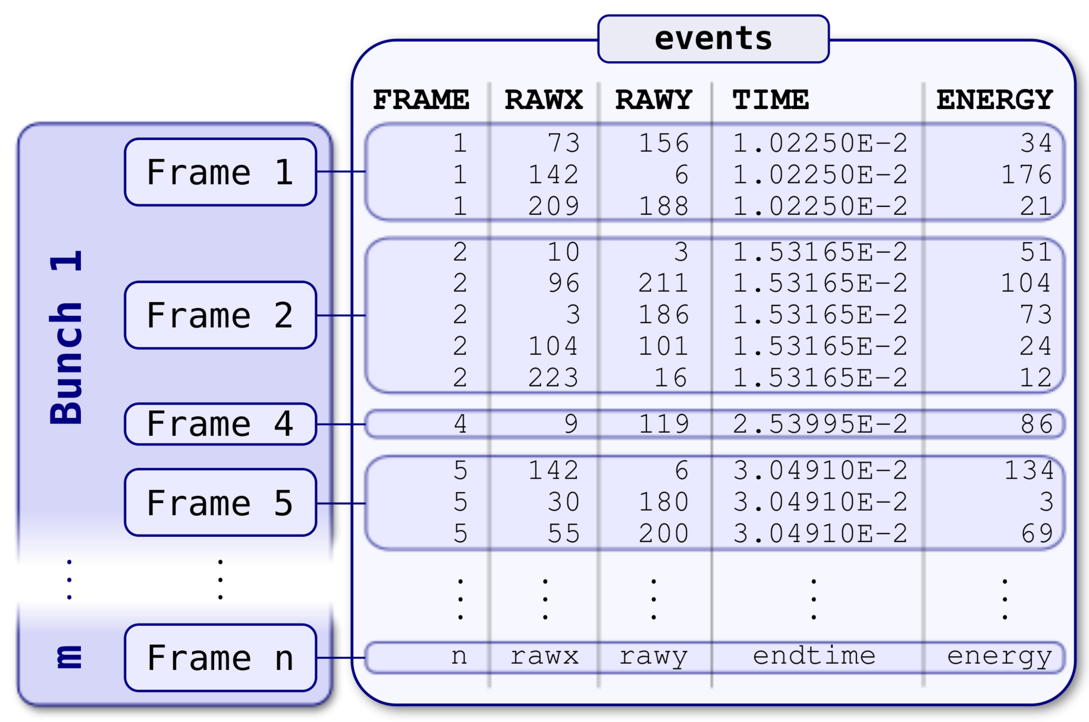
Cutout of an image taken by one of the *XMM-Newton* pn-CCDs. Several hot pixels, a bright line and a presumable source (extended bright spot) are clearly visible.

Hot pixel recognition



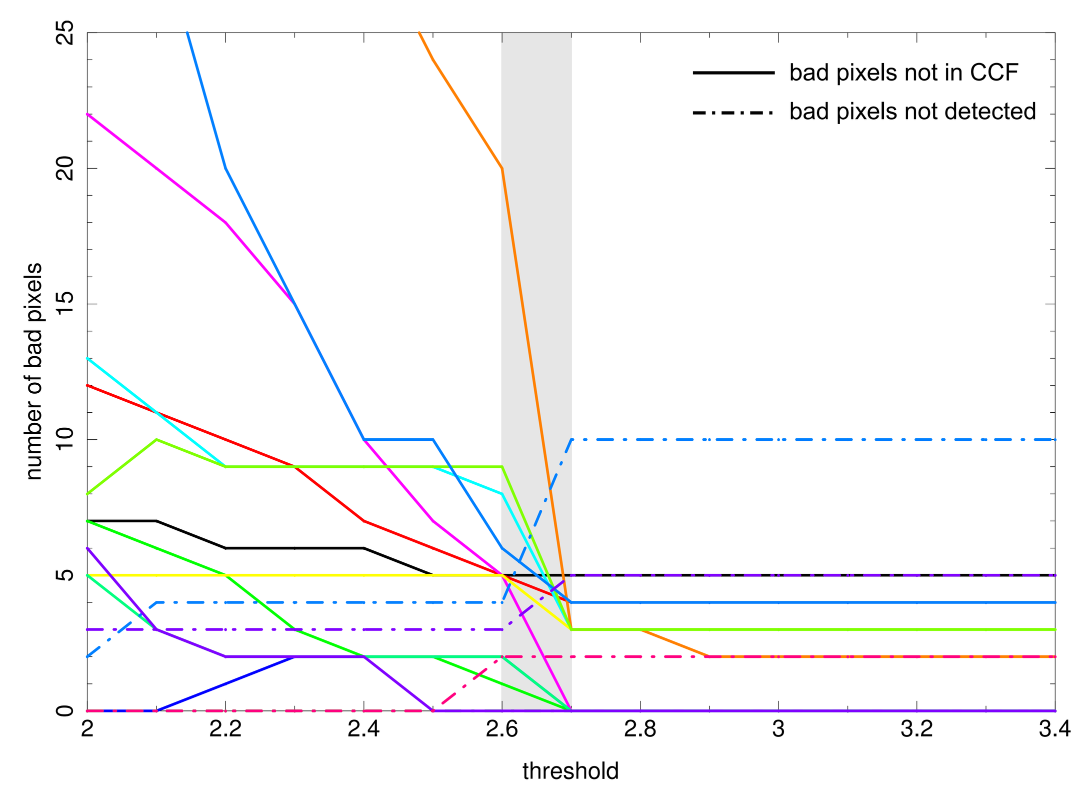
After event preprocessing the bunches of data are split into smaller parts with each of them going through several statistical tests on hot pixels and bright lines. A first check detects the actual presence of bad pixels/lines in the current bunch while consecutive checks filter all findings for relevance by looking at their total abundance.
Two output files in FITS-format will be generated: a bad pixel map and a table which provides more detailed information about each bad pixel finding.

Event preprocessing



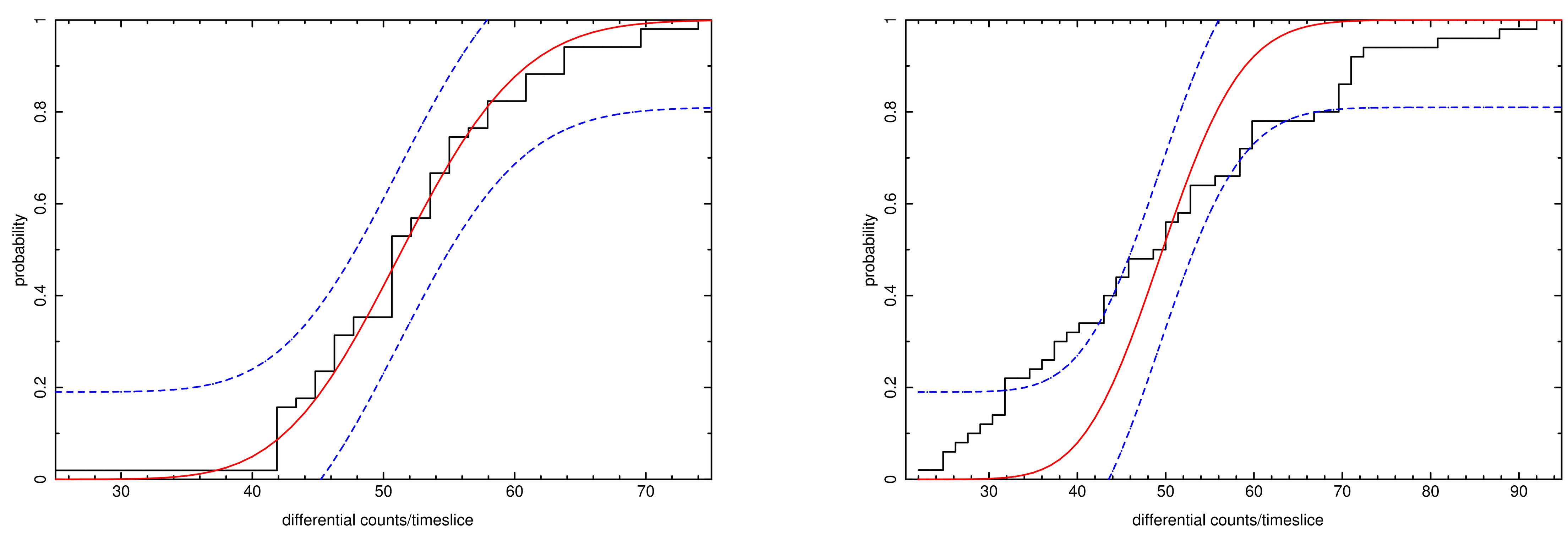
Input data is required to be organized in eventlists which are sorted according to frame or time information. The data is split into equal bunches, each of them containing the cumulated events of several sequent frames. The framepacks are then fed into the recognition algorithm.

Performance tests



The figure shows a comparison of several bad pixel searches on *XMM-Newton* data (different colors are different data sets) with the contents of the corresponding official calibration files.
Solid lines suggest bad pixel findings which were not listed in the calibration files while dashed lines indicate pixels present in the calibration file but not detected by the algorithm. Each search was performed for multiple different detection thresholds. The shaded area marks the region of best agreement with the official bad pixel database.

Tests on constant/flickering nature of hot pixels



KS-tests for two hot pixels: the figures show counts-per-timebin histograms (black line) of a constantly bright pixel which passed the test (left panel) and of a flickering hot pixel. The red lines indicate the cumulative poisson distribution for the respective mean counts while the dashed blue lines mark the boundaries of the 95% confidence level. They serve as a critical value in the decision whether the pixel is flickering or not.
If a pixel shows very many counts per timebin, a χ^2-test is performed instead of a KS-test.

References and Acknowledgements

J. Wilms et al, 2009, "eROSITA Near Real Time Analysis Software Design", eRO-RSB-DD-63-01_2
"XMM-Newton Users Handbook", Issue 2.8.1, 2010 (ESA: XMM-Newton SOC)
M.G.F. Kirsch et al, 2005, "Health and cleanliness of the XMM-Newton science payload since launch", Proc. SPIE, 5898
This research was funded by the BMWi under DLR grant numbers 50 QR 0903, 504 OR 0801

ERLANGEN CENTRE FOR ASTROPARTICLE PHYSICS

FRIEDRICH-ALEXANDER UNIVERSITÄT ERLANGEN-NÜRNBERG