**Constellation School on X-Rays from Star Forming Regions**

**(May 18-22, 2009)**

# PWXDetect: An X-ray source detection code for XMM-EPIC data

*Francesco Damiani and Ignazio Pillitteri*

## 1 Introduction

This document describes the characteristics and the usage of the `PWXDETECT` source detection code developed at *INAF – Osservatorio Astronomico di Palermo "G. S. Vaiana"*, for X-ray imaging data obtained with EPIC onboard the XMM/Newton X-ray Observatory.

The method is based on the analysis of the wavelet transform (WT) of the input image data, and is able to detect efficiently both point-like and moderately extended sources, yielding for each source a probability of existence and accurate position; as a by-product, estimates of the source apparent size and count rate are also provided. The WT is a convolution of the data with a kernel function, depending on position $(x, y)$ and a scale parameter $a$; therefore, it generates a set of images, one for each value of $a$. The convolution kernel is a function $g(x, y, a)$, called *generating wavelet*, for which a wide choice is possible, the best choice depending on the problem to be solved. In the present case, i.e. the search for nearly isotropic, bell-shaped sources over a flat (or anyway smooth) background, the choice has fallen on the so-called *Mexican Hat* function:

$$g(x, y, a) = g(r/a) = (2 - \frac{r^2}{a^2})e^{-r^2/2a^2}$$

with $r^2 = x^2 + y^2$ and where $a$ is the scale parameter (Fig. 1, left). This is isotropic (the same scale $a$ is used for both $x$ and $y$ axes), and bell-shaped, similar to the source PSF. This ensures that the WT has a strong peak at the position of a source, permitting its efficient detection (see Fig. 2). Another important property (common to all generating wavelets) is that its surface mean is zero, and therefore an uniform background is transformed to a zero value on average (fluctuations however remain). This property and the isotropy of $g(r/a)$ ensure that even a background gradient is transformed to zero. The Mexican Hat function satisfies (although only approximately) the other important requirement of generating wavelets of having compact support, i.e. it differs (significantly) from zero only in a finite neighborhood of $r = 0$, say $r \leq 5a$; this is at the origin of the WT ability to detect *local* features in the data (in strong contrast to Fourier transforms).

The scale dependence of the WT causes the intensity of its peak corresponding to a source to depend on the source (apparent) size (Fig. 1, right). Therefore, by

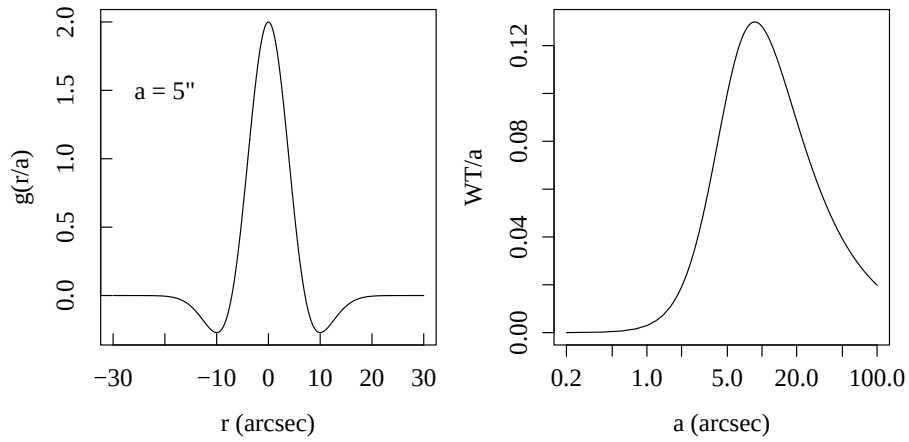


Figure 1: Left: *Mexican Hat* function with a scale parameter of 5". Right: WT/a function for a gaussian PSF with $\sigma = 5$".

varying the scale $a$, PWXDETECT is able to detect sources of different sizes (either intrinsically extended or because of PSF variations).

Despite its insensitivity to background gradients, the WT is however sensitive to second derivatives in the input image data. Therefore, instrumental artifacts such as CCD edges, or gaps between detector chips may easily lead to spurious detections. These features are taken into account in the PWXDETECT implementation.

# 2 PWXDETECT algorithm

An X-ray source detection method based on WT was originally developed at Osservatorio Astronomico di Palermo for the analysis of ROSAT PSPC images (Damiani et al. 1997a,b), and PWXDETECT was later developed along a similar path, to analyze EPIC XMM-Newton data. A similar algorithm (called PWDetect) is also available to work on *Chandra* ACIS and HRC data.

The code runs through the following steps:

1. construction of a background map
2. convolution of image data with the wavelet function at different scales
3. source detection at each scale
4. cross identification of detections at different scales, and buiding of a unique source list
5. update of background image by subtraction of detected sources
6. re-evaluation of detection significance with respect to updated background (repetition of steps 3-4 above)
7. evaluation of detected source properties (count rate, size)

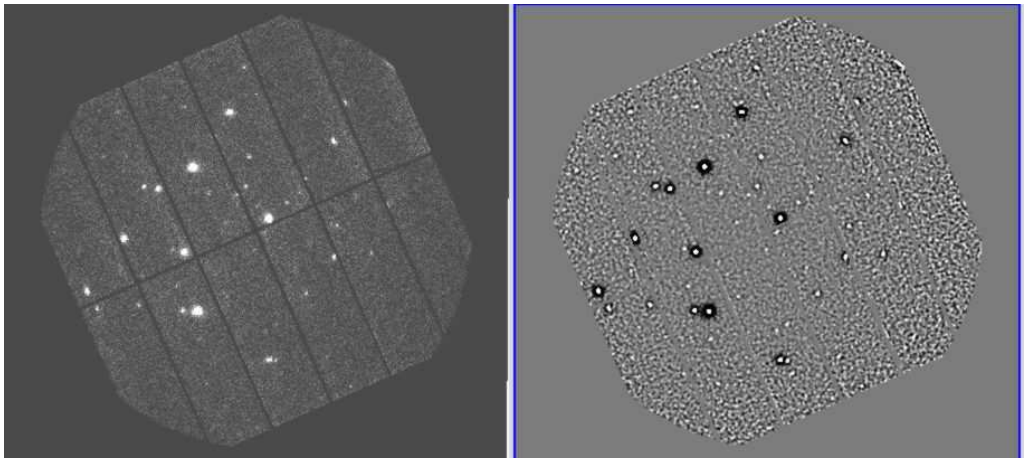

Figure 2: Example of EPIC pn image (left panel) and its Wavelet Transform (right panel).

8. output of the final list of sources.

The algorithm starts with the calculation of a background map, used as a reference to evaluate the significance of the detected sources. Then, the WT image is computed at each scale, and local maxima above a suitable threshold (depending on the local background) are found. A cross-identification is made between sources found at different scale in the same position. The source properties are computed at the scale where the function WT/a reaches its maximum (Fig. 1, right, for the case of a pure gaussian PSF with $\sigma = 5$"). The function WT/a peaks at a scale $a_{max} = \sqrt{3}\sigma$, meaning that the significance of the source is highest when the WT is computed at a scale similar to the source size. If the PSF has a gaussian profile, it is possible to find analytically the relation between the peak of WT at the scale of maximum significance $a_{max}$ and the source counts, thus making possible to derive the source photometry from the WT image, rather than directly from the original image.

In some cases, faint sources near much stronger ones are not detected because the brighter sources cause the computed background to be spuriously high, raising considerably the local threshold. To overcome this problem, sources found in the first iteration are removed from the background map by interpolation (Fig. 3). Then, the list of local maxima found in the first iteration is compared to the new threshold corresponding to the updated, lower background, and a new detection list is built. Thereafter, final, accurate source positions, count rates, and sizes are computed.

# 3 PWXDETECT features specific to EPIC data

In order to work reliably and efficiently on EPIC data, the PWXDETECT code takes into proper account the peculiar features of these data. Among them, most im-

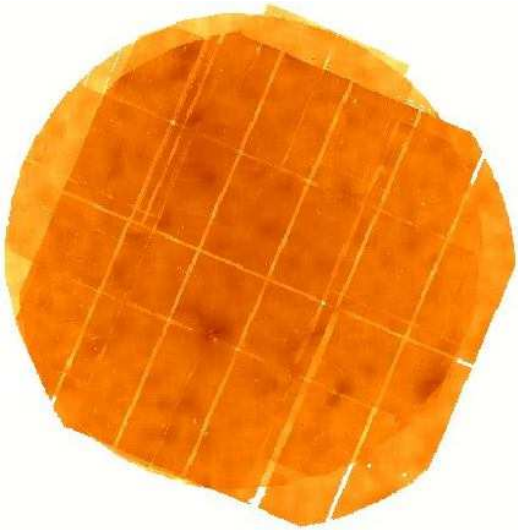

Figure 3: Final background map of a combination of MOS 1, MOS 2, and pn images after source removal.

portant are the shape of the PSF, the shape of image border and gaps between CCD chips, and the simultaneous availability of datasets from three detectors: MOS1, MOS2 (very similar to MOS1) and *pn*.

### 3.1 EPIC PSF

Roughly, the PSF of all three EPIC detectors has a core of size $\sim 5$", but its profile is not gaussian. By calibrations on images obtained after the XMM launch, the PSF has been described by a King profile (Ghizzardi 2001, tech. report EPIC-MCT-TN-011):

$$PSF \propto \frac{1}{(1+(r/r_c)^2)^\alpha}$$

for each of MOS1, MOS2 and *pn* detectors. The $r_c$ and $\alpha$ parameters in the model depend linearly on photon energy and off-axis distance. A remarkable feature of this PSF are its wide wings, so that a relevant fraction of photons fall at relatively large $r$ (for photons with energy $\sim$ 1 keV the Encircled Energy is $\sim$ 80% for a 30" source integration radius, thus 20% is the fraction outside 30"). This is very different from the case of a gaussian PSF, where only a minimal fraction falls outside $3\sigma$ (i.e. three times the core radius). Therefore, we have obtained via numerical integration the PSF-related factors to derive source counts from the WT peak (at the scale of maximum significance). The numerical integration (WT computation) is made, with the same method used in PWXDETECT for total consistence, on a grid of two-dimensional PSF images, at different energies and off-axis angles. The peak value of the WT of the normalized PSF, computed at the scale of maximum significance is the sought-after factor to derive counts from the amplitude of a WT peak, depending on position $(x, y)$, energy, off-axis angle, and detector used (although differences among detectors are not striking).

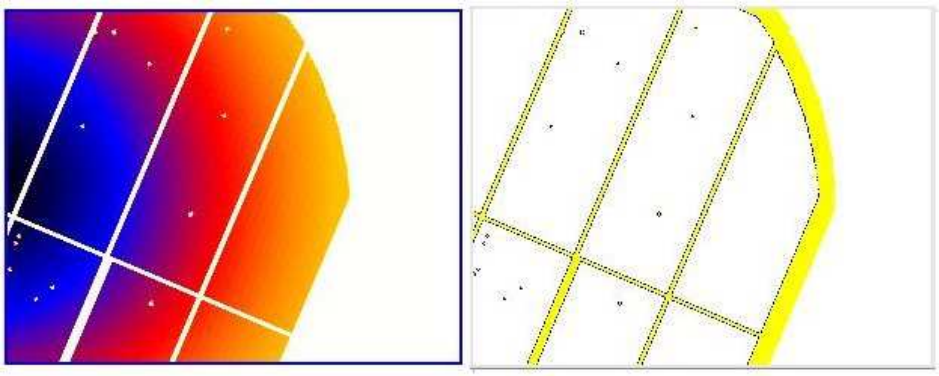

Figure 4: Examples of exposure map (left panel), and extrapolated border (right panel) for the *pn* detector.

### 3.2 CCD Gaps and Edges

Differently from ROSAT and Chandra data, the images obtained with the EPIC camera are characterized by sharp gradients in correspondence to the gaps between adjacent CCD chips, at the chip outer edges and at bad pixel positions. This feature makes a wavelet detection method prone to find many spurious detections in the regions with steep gradients. We have solved the problem with a series of "masks", computed from the exposure maps. By using these masks, PWXDETECT interpolates the real image in the gaps and in the bad-pixel holes, using a weighted average of values from a small neighborhood with non-zero exposure.

Near the image outer edges, the code extrapolates the values of local background outwards by a $80''$. This extrapolated border avoids obtaining local WT maxima (and thus spurious detections) near the true image edge (Fig. 4).

### 3.3 Combination of different MOS and *pn* datasets

The simultaneous availability of three MOS1, MOS2 and *pn* datasets makes possible to perform source detection on the combination of the data, to achieve a deeper sensitivity than using a single instrument. PWXDETECT can combine several images together (not necessarily two MOS and one *pn*), with weight factors given by the exposure time (maps) and the detector effective areas (Fig. 5). The WT is therefore applied to an overall count-rate image rather than on photon images. The difference is not sensible when dealing with individual detectors, but becomes crucial in the case of a combination of datasets, to compensate for the individual detector characteristics (edge, gaps), and take properly full advantage of the information content of all available data.

# 4 Significance threshold of sources and background simulations

Statistical fluctuations of the background in a real image may give rise local maxima in the WT, which might be mistaken as real detections if a detection threshold is not properly set. A detections threshold is expressed as $n\sigma$, where $\sigma$ is the usual standard deviation of a Gaussian distribution; actually, since the WT probability distribution is not a Gaussian when the background is Poissonian, a $n\sigma$ threshold is meant to be equivalent to a probability level (e.g. $3\sigma = 99.73\%$), and we speak therefore of "equivalent sigmas". Pre-setting a threshold in terms of some number of (equivalent) sigmas has only a limited usefulness: a more interesting parameter for thresholding is the expected number of spurious detections in a given PWXDETECT run, which a user is left free to vary according to his/her needs. How this number of spurious detections varies with the $n\sigma$ threshold needed by PWXDETECT has been therefore studied in detail. This was done by running PWXDETECT on large sets (several hundreds) of simulated EPIC images containing only background, as already done for the ROSAT/PSPC detection code (Damiani et al. 1997), and by studying the distribution of significance for all (spurious) detected sources. The exposure map was used, by a purposely written simulator, to produce most realistically background vignetting and inter-chip gaps. Both MOS and *pn* datasets were simulated, and analyzed separately and in combination. We have found that the threshold corresponding to a given number of spurious detections per field increases slightly with the background level, parametrized by the total number of photons in the field of view, and such a dependence was calibrated by analyzing sets of simulations with different total number of counts (for each of MOS and *pn*). The number of simulated datasets at each background level was large enough to permit a reliable derivation of the threshold corresponding to one spurious detection per field. For a given detector (or combination), the threshold is therefore uniquely fixed by the desired (average) number of spurious detections per field, and by the image total background.

Indicatively, the threshold corresponding to to one spurious detections per field is 4.7-4.8 for images with 100 kcounts, 4.9-5.0 for 150-200 kcounts and 5.2 for 300 kcounts, approximately the same for MOS and *pn*. In case of a sum of datasets, the same rule applies, provided that the total number of background counts is taken as reference.

# 5 PWXDETECT usage

`PWXDETECT` is able to detect sources on a single EPIC event dataset, or on several of them with nearly the same pointing. The code needs as input a valid FITS event file (or a list of event files), and its corresponding exposure map (or list of exposure maps). The exposure map must be computed beforehand using the SAS task **eexpmap**, binned (mandatory) at a scale of $2''$ pix$^{-1}$, resulting in a size of 1296×1296 pixels (i.e., a binning of 40 pixels in both X and Y columns of event

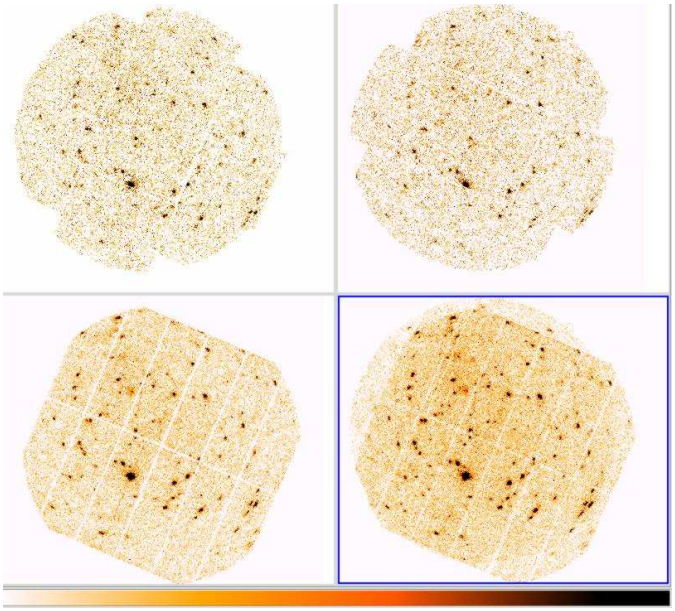

Figure 5: From top to bottom, and from left to right: EPIC MOS1, MOS2, *pn*, and the 'combined' image obtained with PWXDETECT.

tables). Filtering "on the fly" is not possible: any time or energy filtering must be applied before calling `PWXDETECT`.

`PWXDETECT` uses the parameter file *pwxdetect.par* to read/save its parameters. This file must reside in the working directory and its name cannot be changed. Normally, `PWXDETECT` asks interactively for input parameters. Non-interactive mode (for e.g. shell scripts) is set by changing in the *pwxdetect.par* file the *mode* parameter from 'ql' to 'h'. `PWXDETECT` requires two important input parameters: the detection threshold and the maximum scale to be used in the analysis. The number of expected spurious detections depends on the choice of these parameters, and on the image background level (see previous section). The code has been tested mainly on pointlike sources and thus it works with detection scales up to $16''$.

## 5.1 Input parameters and output files

*Input file name*: a standard EPIC FITS event table, or the name of a file (preceded by a '@') containing a list of datasets, one per line.
*Exposure map file name*: the exposure map of the given event file, created with *eexpmap*, or the name of a file (preceded by a '@') containing a list of exposure maps, one per line, in exactly the same order as the listed event files.
*Final significance threshold for detection*: the final detection threshold, in equivalent gaussian sigmas (see above). *initial threshold*: the initial threshold is not asked for in interactive mode, and its value may be changed only by editing the pwxdetect.par file. Based on our experience, we recommend that it should be at most one less than the final threshold (e.g., 3.5 for a final threshold of 4.5), or even slightly

less.
*Maximum wavelet scale*: the input dataset is convolved with a mexican hat wavelet at various spatial scales. The smallest scale is set to 4.0 arcsec to be $\leq$ the EPIC PSF size. The largest scale can be chosen by the user, nominally up to a value of $32''$. The scales actually used by `PWXDETECT` are all powers of $\sqrt{2}$, with other values rounded to the nearest power of $\sqrt{2}$. For point sources the value of the maximum useful wavelet scale is 12-16$''$ depending on the crowdedness of the field. Maximum scales larger than $16''$ are not well tested, and thus not recommended.
*Output source list filename root*: the root file name for detected source lists (and DS9 region file).
*Output background filename root*: the root file name for output background maps.
*Output Log file name*: the file name of a log of `PWXDETECT` run.

The `PWXDETECT` output comprises the following files:

- a list of detected sources (root_src, root_src.fits), in ASCII and FITS format, respectively, containing source properties (position, counts, count rate, size, exposure time, background value, etc., see below).
- an ASCII list which contains for each source and for each scale the detection significance level (root_signif)
- a DS9-style region file (root_overlay.ds9), for visualization of detected sources with DS9 (ftools software suite).
- a log file containing the same output as printed on the screen.
- two FITS background maps: a zero-order map (bkgroot0.fits), and a final map (bkgroot.fits) with bright sources removed. Units are counts/square arcsec.
- an image of count rates (rate_image.fits) in counts/sec/pixel, and a mask.fits file (intermediate products, for checking purposes only).
- in case of a sum of images, the following outputs are created: a list of events with X and Y coordinates (merged_evtlist.fits), and a normalized, summed exposure map (merged_expmap.fits),
- Optionally, a sensitivity map can be created after the detection process. This option is enabled during code compilation, and thus depends on the code version being used. Note: this single step requires normally much more run time than the usual detection run.

**Sum of data: Inputs and outputs:**
In the case of a sum, a list of datasets is processed. The detector of the first dataset is used as reference instrument and the other datasets are scaled (through factors input by the user) to this reference detector.

The input file names must be preceded by a "@" symbol (e.g.:"@maplist" to read in the *maplist* file); in this way the code reads the list files of event tables

and maps. To combine several dataset (for example MOS1, 2 and pn) the exposure map list file must contain two numbers after each exposure map file name (in the same line): a number proportional to the respective background levels in the input datasets, important to ensure that no unexpected spurious detections are obtained in combining them; and an effective area (in $cm^2$) for each detector (either monochromatic or spectrum-weighted, as chosen by the user), used to scale the count rates to the reference detector, to derive the X-ray photon flux for detected sources. The first number can be the total number of events in each dataset (assuming that background counts are much more than total source counts, otherwise it should be evaluated more accurately). The second number should be the mean effective area of each detector, weighted with the spectrum of sources that are expected to be found (or those that one aims to study with more accuracy).

Exposure times listed in the output files are the simple sum of exposure times of each map, while the rates and counts are calculated by taking into account the weighted sum of the different detector responses and vignetting. They are scaled to the reference detector, hence to derive the energy fluxes (ergs $cm^{-2}$ $s^{-1}$) the conversion factor between count rates and energy fluxes must be calculated for that reference detector and for the corresponding filter used during that observation. Obviously, changing the order of the listed datasets changes the reference detector. The nominal pointing direction of the combined image merged_evtlist.fits is the same as the input image with the highest count statistics.

## 5.2 The output file *root*_src

The detected source file *root*_src[.fits] contains:

Column # 1: `RA` (deg): Right Ascension (decimal degrees)
Column # 2: `RA_err` (deg): Error on RA (decimal degrees)
Column # 3: `Dec` (deg): Declination (decimal degrees)
Column # 4: `Dec_err` (deg): Error on Dec (decimal degrees)
Column # 5: `X` (pix): Position X (physical pixels)
Column # 6: `X_err` (pix): Error on X (physical pixels)
Column # 7: `Y` (pix): Position Y (physical pixels)
Column # 8: `Y_err` (pix): Error on Y (physical pixels)
Column # 9: `Offax` (arcmin): Off-axis angle
Column #10: `Det_scale` (arcsec): Highest-significance detection scale
Column #11: `Src_area` ($arcsec^2$): Approximate source (core) area
Column #12: `Signif`: Detection significance (sigmas)
Column #13: `Src_cnt`: Total source counts
Column #14: `Cnt_err`: Error on source counts
Column #15: `Bkg_cnt`: Background counts in source core area
Column #16: `Src_ct_rate` (cts/sec): Source count rate
Column #17: `Src_rate_err` (cts/sec): Error on source count rate
Column #18: `Src_flux` (cts/sec/$cm^2$): Source photon flux
Column #19: `Src_flux_err` (cts/sec/$cm^2$): Error on source flux
Column #20: `Bkg_rate` (cts/sec/$arcsec^2$): Background count rate

Column #21: `Exp_time` (sec): Source size-weighted exposure time
Column #22: `Size` (arcsec): Source size
Column #23: `Siz_err` (arcsec): Error on source size
Column #24: `Extent`: Source extent (relative to PSF)
Column #25: `Ext_err`: Error on source extent
Column #26: `Src_num`: `PWXDETECT` source number

# 6 Preparation of data

This section describes how to filter the data in order to maximize the ability to detect faint sources by reducing the background level and spurious artifacts in EPIC images. The data can be obtained by the Pipeline Processed System (PPS files) or by having to rerun the reduction chains for MOS and pn starting from Observation Data Files (ODF files). Details on the general processing of EPIC data can be found in the document at the web address: `http://heasarc.gsfc.nasa.gov/docs/xmm/abc/`. We suppose to start from event tables generated by the SAS pipeline reduction software or retrieved from PPS files. We give here further details about energy, pattern and time filtering.

Generally, the data from the standard SAS pipeline reduction need to be filtered for energy band, pixel triggering pattern and time intervals with low background rate. The SAS task **evselect** is used throughout these steps. An example of **em evselect** command is the following:

```
evselect \
--table='P0041750101PNS002PIEVLI0000.FIT' \
--withfilteredset=true \
--filteredset=filteredPN_03.79.fits \
--keepfilteroutput=true \
--destruct=yes \
--withimageset=false \
--writedss=true \
expression="!( (CCDNR.eq.5) .and. (RAWX.eq.11)) .and. ((FLAG & 0xfb0024)==0) \
.and. (PATTERN .LE. 12) .and. (PI.ge.300) .and. (PI.le.7900)
```

This command will filter the event table P0041750101PNS002PIEVLI0000.FIT to obtain the filtered file filteredPN_03.79.fits with the criteria given in the *expression* parameter. The expression operates a selection on FLAG and PATTERN parameters, energy band and excludes a group of pixels.

The filters which should be applied to the initial data are described below.

**Pattern selection.** A "pattern" flag is assigned to each event in order to qualify the manner in which neighbour detector pixels have been triggered simultaneously by each X-ray photon. A convenient choice is to retain only the events that triggered up to four pixels at the same time (in fortran-like syntax: `PATTERN .le. 12`). For spectral analysis a restrictive choice is `PATTERN .le. 4` to retain only double pixel events.

**Energy filter.** This filter selects the events in a certain energy band through the PI column (PI units are eV).

**FLAG filter.** In Fig. 6, left panel, the raw image is affected by spurious pseudo-events near the chip edges and by hot pixels. Such features could result in many spurious detected sources. It is of particular relevance to clean the image from these effects if we are interested to obtain a list of detected sources almost free from false detections. It is possible to filter the event by means of FLAG column in MOS and pn. The expression to be used is written in the keyword XMMEA_EM and XMMEA_EP in the header of event FITS table for MOS and pn respectively. While for MOS data the filtering with this expression (namely: `FLAG & 0x766b0000)==0`) is adeguate, for pn data the analog expression: `FLAG & 0xfa0000) == 0` is not. The various flag listed in the header keywords correspond to expressions given in exadecimal notation. The (exadecimal) sum of two values performs the combined action of the summed flag filters. The expression `FLAG & 0xfb0024)==0)` makes the above pn selection and in addiction excludes the events: out of fov, close to chip gaps, close to bad pixels; namely, the sum of exadecimal values: (0x)fa0000 + (0x)4 + (0x)20 + (0x)010000. The right panel in Fig. 6 shows the result of these filters.

**Bad pixels.** It is possible to exclude group of pixels or entire chip columns. In the above example the section !((CCDNR.eq.5) .and. (RAWX.eq.11)) exclude the damaged column 11 of chip 5. This bad column is the bright one in the top left corner of raw image in Fig. 6, left panel.

### 6.1 Time filtering

XMM-Newton observations are very often affected by high background rate intervals. Because of the high satellite effective area, during high solar activity episodes, high energy particles are collected by the XMM-Newton mirrors and produce contaminant events. We discuss in more detail this filtering stage and a method based on the signal to noise of the image.

## 7 References

The algorithm and the application to the PSPC detector are described in:
– Damiani et al. 1997, ApJ, 483, 350
– Damiani et al. 1997, ApJ, 483, 370
For technical and calibration information about EPIC and XMM-Newton:
`http://xmm.vilspa.esa.es`

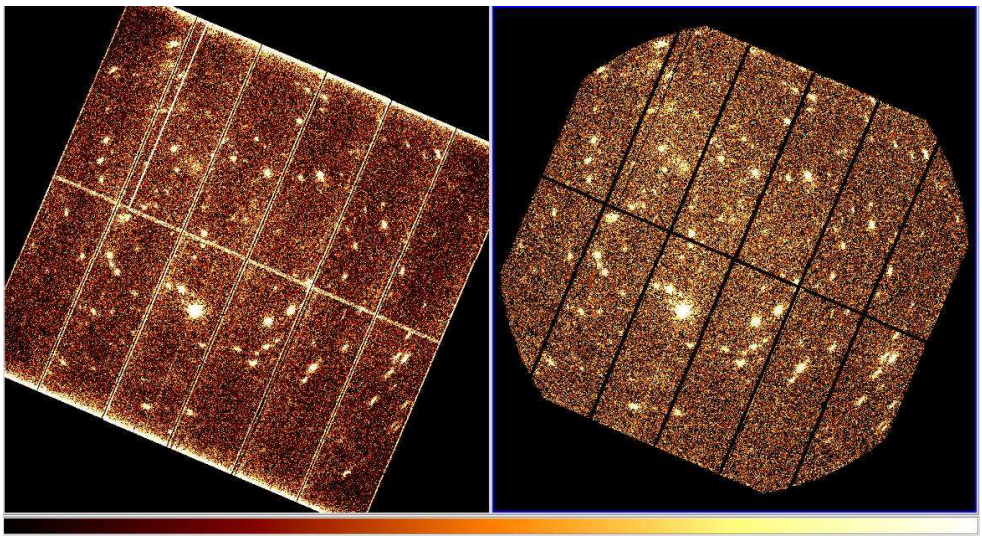

Figure 6: Left panel: pn data from SAS reduction pipeline. Right panel: the same data filtered for energy, flag, pattern and low background rate intervals.