epevents

August 16, 2005

Abstract

This task converts an EPIC PN raw events file (produced by **epframes** and **badpix**) into a calibrated event list with flagged trailing events and corrected for gain variations of the preamplifier channels and charge transfer inefficiency (CTI). It performs a pattern analysis to identify and recombine split events and converts the raw unlinearized detector coordinates into corresponding physical locations.

1 Instruments/Modes

Instrument	Mode
EPIC PN	IMAGING, TIMING, BURST

2 Use

pipeline processing	yes
interactive analysis	yes

3 Description

3.1 General

The EPIC PN raw event list, processed by **epframes** and **badpix** is used as input.

epevents flags all events with a gain corrected amplitude smaller than a parameterized value which follow in the same CCD column immediately after an event with an amplitude above a certain (fixed) threshold as 'trailing' events. Such artificial events are produced by re-emission during the readout process ([1]). The flags are bit-masked according to [3]. The flagging is suppressed for TIMING mode data. In this mode the reemission effect is very unlikely as events are binned into macro pixels.

A pattern analysis is made which yields four distinctive parameters: the number of pixels in the structure, the running pattern number within a frame, the pattern orientation (zero for singles and unresolved patterns; otherwise a three digit code, with the last digit specifying the orientation of the pattern and the first two the position of the pixel on the pattern) and the distance (in pixels) to neighbouring event

in the same frame and column along the readout direction towards the CAMEX (0 indicates the first pixel read in a column). All four parameters are combined into three output columns PAT_ID, PAT_SEQ, and PATTERN which will be part of the EPIC pn calibrated event list as a PPS product ([2]). Bad pixels from the BADPIX extension are not included in the pattern analysis.

Events are flagged as on a bad pixel if they coincide with a bad pixel in the BADPIX extension. Single events (or patterns, based on their pixel with maximum amplitude) next to a hot or bad pixel are flagged accordingly.

Detection of cosmic ray events (MIPs) is based on the high energy cutoff. Depending on the MIP rejection mode (on-board, in **epframes**) information from the discarded line information in the EXPOSURE extension is used for event flagging.

Making use of calibration products in the CCF and CAL routines, **epevents** corrects raw energies (PHA) for gain variations of the preamplifier channels and CTI during readout to compute the PI values. The CTI correction is calculated as a function of column position, (temperature, if requested), PHA of the events as well as distance and PHA of the preceding event along the readout direction. Hence, the calibration procedure requires information on events read out from the same node prior to the event in question (memory-effect).

epevents recombines all the events split across several pixels with the same pattern index: the energy of the recombined event is the sum over the split energies, its position is that of the split event with maximum raw energy. There are $\sim 70\%$ single events and $\simeq 30\%$ double patterns (other valid events are triples and quadruples), the exact numbers vary with energy. **epatplot** is a tool to visualize the pattern type distributions as function of (recombined) photon energy.

Finally, **epevents** computes physical camera detector coordinates DETX, DETY (in the CAMCOORD2 system) from raw detector coordinates RAWX, RAWY which may be randomized over the pixel area depending on the parameter randomizeposition.

For readout modes TIMING and BURST the source position attribute SRCPOS written by **epframes** is used to create a DETY coordinate corresponding to the source position on the CCD. SRCPOS is also used as position for the CTI correction.

3.2 Out-of-time events

3.2.1 General

Depending on the observation mode a certain fraction of the events recorded are due to out-of-time events (OoTE), i.e. are registered during readout of a CCD and thus will be assigned an incorrect position in RAWY and consequently an incorrect CTI correction. Typical values are 6.3% for full frame mode, and 2.3% for extended full frame mode (with FRAME_TIME_PARAMETER=3).

Out-of-time events have a much lower exposure per pixel per read-oiut frame, namely reduced by the number of pixel shifts during readout and by the abovementioned ratio of readout and cycle time. Therefore the likelihood for pile-up is extremely reduced (by more than 3 orders of magnitude). Extremely bright (and hard) sources can trigger the MIP rejection mechanism (see **epframes**) due to pile-up: if 2 (or more) photons hit the same pixel during one read-out frame the accumulated charge may exceed the limiting value of 3000 adu. Thus this "event" will be treated as MIP and this CCD column and the neighbouring ones get rejected in this read-out frame. This causes a spatial inhomogeneity in the exposure (a warning is issued by **epframes** in this case), and especially the number of recorded out-of-time events is significantly reduced in CCD columns in the core of the PSF (where they are usually strongest).

3.2.2 Analysis of extended sources or in the vicinity of moderately bright sources

If `withoutoftime`=y the task does *not* create the "normal" events file but uses the input file instead to create an event file where all events are *treated* as out-of-time events. After the pattern recognition for the same `TIME`, `PHA`, and `RAWX` a new `RAWY` is simulated by randomly shifting the pattern along the RAWY axis and performing the gain and CTI correction afterwards.

This out-of-time events file has the same temporal variations and pattern distribution as the "normal" event file. All events have an energy correction corresponding to the (randomly distributed) RAWY column. You can apply the same selection on both the "normal" and "out-of-time" event file, scale with a mode-dependent factor and subtract the "out-of-time" contribution from the other. You can create "out-of-time"-corrected spectra and images for an arbitrary time and energy range by subtracting the scaled product derived from the out-of-time events file from the product created from the "normal" events file.

The current (experimental) implementation does not automatically detect the instrument and bad pixel setting. The `RAWY` range to distribute the out-of-time events is instead derived from the `RAWY` range of the actual events in the data. The mode-dependent conversion factor has thus to be applied manually and is about 95% of the out-of-time event fraction (e.g. 0.060 for full frame mode) due to setting the top 10 rows to "bad". This will we improved in later versions.

For an example how to use the current implementation of out-of-time event handling in an event file processing, consult the documentation of **epchain**.

Note that this parameter is only meaningful for IMAGING modes. Moreover, in the *Large Window* and *Small Window* modes out-of-time events can be introduced by sources that are outside the readout-window. These cannot be accounted for. Similarly, for *Full Frame* and *Extended Full Frame* modes (currently) the 11 rows closest to the CAMEX are set bad onboard but sources in these rows may also cause unrecognized out-of-time events. Finally, if a source is highly affected by pile-up this feature is projected onto the distribution of out-of-time events as well, however, the actual recorded out-of-time events do not suffer from this and thus residual effects remain.

3.2.3 Analysis of very bright sources

A different approach is done via the setting `withctisrcpos`=Y: all event energy corrections do not use the actual RAWY coordinate but the corresponding RAWY location of the source (`SRCPOS`) as determined by the task **epframes** (either automatically or via setting `withsrccoords`=Y). So all events are assumed to originate from the source position. Events that really occur at a RAWY different from `SRCPOS` get a wrong (usually too high) CTI correction instead. (Note, that this procedure is the correct handling for TIMING and BURST mode – where it is performed automatically). A spatial selection should be applied in the final event analysis to deal only with detector portions dominated by Out-of-Time events. The task **epchain** may be used with the setting `ccds`=4 `withctisrcpos`=Y to process only the CCD 4 where usually the bright source is located that causes the Out-of-Time events.

3.3 Examples

Let us assume we have an output event file produced by **epframes** and **badpix** with name `rawevents_04.dat`, then the simplest call to **epevents** is

```
epevents eventset=rawevents_04.dat outset=events_04.dat
```

If you want to produce a diagnostic file containing CCD maps in 4 energy bands in the same run then you may use

```
epevents eventset=rawevents_04.dat outset=events_04.dat \
  withphotonmap=Y photonmapset=eventmap_04.dat          \
  lothresh="0 200 500 0" hithresh="200 500 2000 32000"  \
  mappatterntype=sssn
```

This creates three maps with single events only ($0 - 0.2\,\mathrm{keV}$, $0.2 - 0.5\,\mathrm{keV}$, $0.5 - 2.0\,\mathrm{keV}$) and one with all non-single events in the energy range $0 - 32\,\mathrm{keV}$. The parameters `lothresh`, `hithresh`, and `mappatterntype` are used with default values. A non-default call is

```
epevents eventset=rawevents_04.dat outset=events_04.dat \
  withphotonmap=Y photonmapset=eventmap_04.dat          \
  lothresh="0 0 0 0" hithresh="32000 32000 32000 32000"  \
  mappatterntype=sxvw
```

which creates one single events map, one with doubles perpendicular to the readout direction, and two maps with doubles in readout direction where one has the main event near the CAMEX and the other with the main away from the CAMEX.

4 Parameters

This section documents the parameters recognized by this task (if any).

Parameter	Mand	Type	Default	Constraints
eventset	yes	data-set		

EPIC PN raw events data-set

outset	yes	data-set	events.dat	

EPIC PN events data-set with calibrated and corrected PI channels

reemissionthresh	no	integer	0	none

selection parameter: trigger threshold (in adu) for preceding events

randomizeposition	no	boolean	Y	Y/N

yes, if the computation of physical camera detector coordinates is done with randomization (*former parameter 'randomize'*)

randomizeenergy	no	boolean	Y	Y/N

yes, if the raw amplitudes should be randomized within a pulse-height bin

testenergywidth	no	boolean	N	Y/N

yes, if use non-standard energy bin width, this will yield *incorrect scientific results* and should *only* be used for calibration tests, no support

gainctiaccuracy	no	integer	2	0-2

Accuracy of gain/cti correction

withphagaincolumn	no	boolean	N	Y/N

Whether to create intermediate column `PHA_GAIN`.

lowgainenergyscale	no	boolean	N	Y/N

When switching on the energy correction for low-gain mode data then most of the events will fall outside the 2-byte-limit for the `PI` column (i.e. $> 32757\,\mathrm{eV}$) as the energy range is then about $2 - 280\,\mathrm{keV}$; if one is interested in this full range the setting "N" should be used and energy values be multiplied later with 18.4 to obtain "real" event energies. Only effective for the few low-gain mode exposures, of course.

patternanalysis	no	boolean	Y	Y/N

no, if pattern recognition has been done already (future development, not active yet)

withphotonmap	no	boolean	N	Y/N

yes, if diagnostic photon map file should be created (see parameter `photonmapset`)

photonmapset	no	data-set	photonmap.dat	

name of optional output diagnostic photon map file, it will have 4 extensions corresponding to 4 user-definable energy bands (see `lothresh` and `hithresh`)

lothresh	no	real-list	0 200 500 0	

lower band limits [eV] for parameter photonmapset

hithresh	no	real-list	200 500 2000 32000	

upper band limits [eV] for parameter photonmapset

mappatterntype	no	string	sssn	

pattern types for the events of the individual band maps (future development, not fully active yet), s =

singles, d = doubles, t = triples, q = quadruples, n = non-singles, r = recognized, u = unrecognized, m = multiples (m = d+t+q), a = all, x = doubles in x-direction, y = doubles in y-direction, v = doubles in y-direction with main near CAMEX (i.e. split backward), w = doubles in y-direction with main away from CAMEX (i.e. split forward)

withoutoftime	no	boolean	N	Y/N

yes, if "out-of-time events" file should be created *instead* of "normal events" file (only meaningful for IMAGING modes) (considered as *experimental*)

withctisrcpos	no	boolean	N	Y/N

yes, if not the RAWY coordinates but the source position SRCPOS should be used in the energry correction routines (only meaningful for IMAGING modes) (considered as *experimental*). See Sect.3.2.

withctilongterm	no	boolean	Y	Y/N

yes, if corrections for long-term CTI increase should be applied

ctilongtermsoft	no	boolean	N	Y/N

yes, if special soft energy function should be included in the long-term CTI corrections (considered as *experimental*)

ctilongtermy	no	boolean	Y	Y/N

yes, if special Y-dependence should be included in the long-term CTI corrections

withccdoffsets	no	boolean	N	Y/N

yes, if CCD offset corrections should be applied (considered as *experimental*)

withtempcorrection	no	boolean	N	Y/N

yes, if temperature-gain corrections should be applied (considered as *experimental*)

withframecti	no	boolean	N	Y/N

yes, if TIME-derived frame numbers should be used in CTI correction for non-imaging modes (TI, BU) instead of the ODF frame numbers. For FF, eFF, LW, SW modes internally always the TIME-derived frame numbers instead of the dummy ODF numbers are used (should not be changed).

withgainburst	no	boolean	Y	Y/N

apply special gain if BURST mode ?

withgaintiming	no	boolean	Y	Y/N

apply special gain if TIMING mode ?

withgainff	no	boolean	N	Y/N

apply special gain if FULL FRAME mode ?

withgaineff	no	boolean	Y	Y/N

apply special gain if EXTENDED FULL FRAME mode ?

5 Errors

This section documents warnings and errors generated by this task (if any). Note that warnings and errors can also be generated in the SAS infrastructure libraries, in which case they would not be documented here. Refer to the index of all errors and warnings available in the HTML version of the SAS documentation.

noEvents *(error)*
No `EVENTS` extension in input data set

noExposure *(error)*
No `EXPOSURE` extension in input data set

badMode *(error)*
Input data set not recognized as IMAGING, TIMING or BURST mode

badmipMode *(error)*
Mip Mode not supported

tooManyPattern *(error)*
Number of patterns exceeds the limit

tooManySelected *(error)*
Number of selected pattern pixels exceeds the limit

tooManyClasses *(error)*
Number of equivalence classes exceeds the limit

tooManyClassentries *(error)*
Number of entries in event class exceeds the limit

invalidQuad *(error)*
Invalid quadrant identifier

invalidCCD *(error)*
Invalid CCD identifier

noBadpix *(warning)*
No BADPIX extension in input data set
corrective action: -

badEvent *(warning)*
Event is on an uploaded bad pixel
corrective action: continue, the CIF entry for EPN_BAXPIX may be incorrect

CCDlowGain *(warning)*
HK values of this CCD indicate that the mode is not HIGH gain
corrective action: continue, energy scale will be incorrect

unusualQual *(warning)*
Event quality not recognized anymore by new SAS event attributes
corrective action: reprocess raw events list with new version of **epframes**

noGainTemperature *(warning)*
input raw events list does not contain meaningful temperature
corrective action: verify temperature value or use withtempcorrection=N

Twidth *(warning)*
requested non-standard energy bin width in the event amplitude correction, will yield incorrect scientific results ;-)
corrective action: do not publish these spectra, they contain artifacts ;-)

frameCTI *(warning)*
setting withframecti=Y for non-IMAGING modes
corrective action: use this eventfile only for engineering purposes, lowest energies suffer from spectral broadening

veryManyClasses *(warning)*
Number of equivalence classes exceeds half the number of events
corrective action: continue, very large pattern identified after event class analysis

6 Input Files

1. EPIC PN raw events list (contains TIME, FRAME, RAWX, RAWY, PHA, CCDNR and FLAG for one CCD) [from **epframes**] with EXPOSURE extension [created by **epframes**] and BADPIX extension [if pre-processed by **badpix**]

7 Output Files

7.1 Files and columns

1. EPIC PN event file with recombined events corrected for gain and CTI effects. The event file consists of:
   - an EVENTS extension with columns TIME, FRAME, RAWX, RAWY, DETX, DETY, PHA, PI, CCDNR, FLAG, PAT_ID, PAT_SEQ and PATTERN (temporarily we also store detailed pattern informations as additional columns PAT_TYP, PAT_IND and PAT_ORI; columns PHA_CTI list the CTI and gain corrected PHA values of each event prior to the split event recombination (to check corresponding CAL functionality))

- Conversion of detector coordinates to physical camera detector coordinates: For TIMING and BURST mode data the position of the source (given as keyword `SRCPOS` in `RAWY` pixel coordinates as written by **epframes**) is used to calculate the `DETY` value at the source position.
- a `BADPIX` extension (copied if available from input file)
- an `EXPOSURE` extension (copied from input file) with new attributes summarizing the pattern analysis: `ANALYSED` (number of analysed events), `SINGLES` (number of single events), `DOUBLES` (number of double events), `TRIPLES` (number of triple events), `QUADRUPL` (number of quadruple events), `NOTRECEV` (number of not recognized events), `NOTRECPA` (number of not recognized patterns), `MAXPAT` (maximum pattern size), `TRAILING` (number of trailing events), `MIPS` (number of MIPs found) and `RECPHOTO` (number of recognized photons)

2. If `withoutoftime`=Y then *instead* a file is created with the same structure as above but consisting of only out-of-time events (*instead* of the normal events file).

3. If `withphotonmap`=Y then additionally a file `photonmapset` is created containing 4 image extensions according to pattern types and energy ranges specified in `mappatterntype`, `lothresh` and `hithresh`.

7.2 Pattern related information

The following table may serve as a quick reference to select certain pattern types of *recognized events* (i.e. away from CCD edges, bad pixels etc.):

PATTERN	Meaning
0	singles
1	doubles in Y with Y(main)<Y(secondary)
2	doubles in X with X(main)<X(secondary)
3	doubles in Y with Y(main)>Y(secondary)
4	doubles in X with X(main)>X(secondary)
5–8	triples
9–12	quadruples
128	singles at CCD window (RAWX=1, RAWX=64, RAWY=200) or close to bad pixels
205	doubles at CCD window or bad pixels
206	triples at CCD window or bad pixels
207	quadruples at CCD window or bad pixels

Note: as of version 6.30.4 PATTERN values of 128 have been changed to 0 (i.e. 8^{th} bit is not set anymore for singles), and PATTERN values of 205 have been changed to 1–4 (i.e. 7^{th} and 8^{th} bit are not set anymore for doubles)!

Secondary events of those valid doubles, triples, and quadruples (`PATTERN` = 1..12) have PATTERN(main)+64 (as listed above), i.e. 7^{th} bit set.

1. For the pattern codes in `PAT_ID` and `PATTERN` the following bit-wise storing is used:

PAT_ID

bit	value	Meaning
16	32768	free for additional pattern related flag
15	16384	– " –
14	8192	– " –
13	4096	PAT_ORI first digit (x-coordinate)
12	2048	– " –
11	1024	PAT_ORI second digit (y-coordinate)
10	512	– " –
9	256	PAT_IND: $1, ..., < 512$ (telemetry limit)
...	...	– " –
1	1	– " –

PATTERN

bit	value	Meaning	
8	128	sign of PAT_TYP	
7	64	sign of PAT_IND	
6	32		used to flag PAT_TYP > 4
5	16		*if bit 6, then use next 5 bits*
4	8	MOS code numbers 0 - 12	*to store PAT_TYP - 5*
3	4	*a combination of*	*hence, max storage: PAT_TYP = 36*
2	2	*PAT_TYP* ≤ 4 *and*	– " –
1	1	*PAT_ORI*	– " –

Note: as of version 6.30.4 PATTERN values of 128 have been changed to 0 (i.e. 8^{th} bit is not set anymore for singles), and PATTERN values of 205 have been changed to 1–4 (i.e. 7^{th} and 8^{th} bit are not set anymore for doubles)!

2. Creation of event quality flags in column FLAG. **epevents** makes use of the common MOS/pn event related flag code (see [3]) and uses the following bits (other flags are set by the **epframes** task):

FLAG

bit	value	Meaning (information)
1	0x2	INVALID_PATTERN
2	0x4	CLOSE_TO_CCD_WINDOW
5	0x20	CLOSE_TO_ONBOARD_BADPIX
6	0x40	CLOSE_TO_BRIGHTPIX (not on-board)
8	0x100	CLOSE_TO_DEADPIX (not on-board)
16	0x10000	OUT_OF_FOV

bit	value	Meaning (rejection)
19	0x80000	COSMIC_RAY
21	0x200000	ON_BADPIX
22	0x400000	SECONDARY
23	0x800000	TRAILING
total	0xfa0000	EPN rejection mask

7.3 Details about the pattern recognition

epevents performs a pattern analysis to identify and characterize split events: the charge distribution in split events is analysed and the detected patterns are described by four parameters:

1. PAT_TYP: PATTERN TYPE, the absolute value identifies the pattern, i.e. the number of pixels which belong to the pattern: single (1), double (2), triple (3), quadruple (4), ...

negative if amplitude of the corresponding pixel should be ignored (e.g. if not main pixel or or also main pixel if it is at an edge). NULL value is set for trailing events which are excluded from the pattern analysis and MIPs.

2. PAT_IND: PATTERN INDEX, running number of pattern within a frame, starting with 1. All pixels within a pattern share the same (absolute value of) PAT_IND so this parameter is needed to identify split partners. For the pixel of a pattern with the highest PHA this value is positive; for the others it is negative. Unresolved patterns (e.g. those with PAT_TYP > 4) contain only negative values. Singles are not regarded as patterns and thus have PAT_IND=0.

3. PAT_ORI: PATTERN ORIENTATION, zero for singles and unresolved patterns; otherwise for valid patterns a three-digit code, with the last digit specifying the orientation of the pattern and the first two the position of the pixel in the pattern; the pattern is inserted into a 3×3 pixel array with the main pixel (having the highest PHA value) at the center (2,2):

   - first digit: x position of the pixel on the pattern
   - second digit: y position of the pixel on the pattern
   - third digit: orientation of the pattern:
     - for doubles: neighbouring pixel (i.e. pixel with smaller PHA than the main pixel) to the right (1), bottom (2), left (3), top (4) with respect to the local maximum in the pattern
     - for triples: L-shaped distribution with the line of symmetry towards: lower right (1), lower left (2), upper left (3), upper right (4)
     - for quadruples: square-shaped distribution, offset towards: lower right (1), lower left (2), upper left (3), upper right (4)

   All coordinates refer to a chip-oriented coordinate system (ix,iy) with the CAMEX at bottom, seen from the X-ray source, i.e. the x coordinate is running towards the right and the y coordinate upwards, with the CAMEX at y = 0.

4. PAT_SEQ: PATTERN SEQUENCE, distance (in pixels) to neighbouring event in the same frame and column (along the readout direction) towards the CAMEX. PAT_SEQ $= 0$ indicates the first pixel read in a column. This information is used to determine the CTI as a function of the frame history - the charge shift during the readout of earlier events effects the CTI of the actual event.

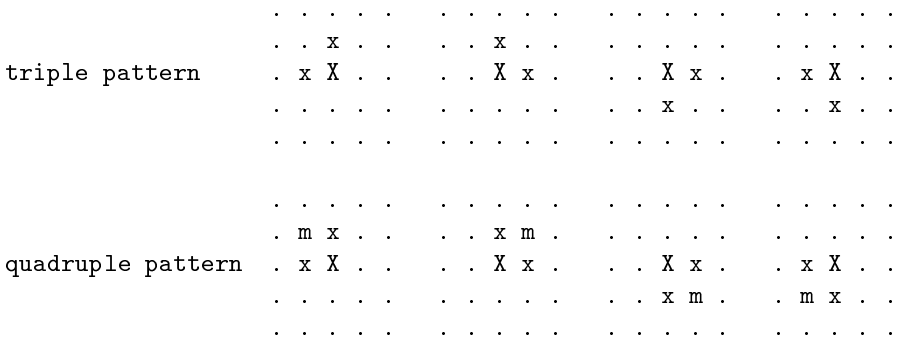
7.4 Classification of the patterns

7.4.1 Single events and recognizable patterns:

7.4.1.1 Isolated case:

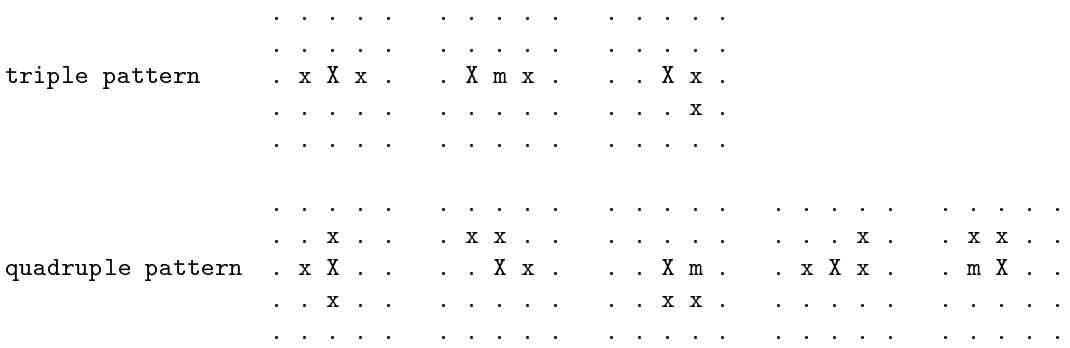
```
                    . . .
single event        . X .
                    . . .

                  . . . . .   . . . . .   . . . . .   . . . . .
                  . . x . .   . . . . .   . . . . .   . . . . .
double pattern    . . X . .   . . X x .   . . X . .   . x X . .
                  . . . . .   . . . . .   . . x . .   . . . . .
                  . . . . .   . . . . .   . . . . .   . . . . .
```



Here '.' marks a pixel without an event above the threshold, 'X' is the pixel with the maximum charge ("main pixel") 'x' is the pixel with a non-maximum charge, 'm' is the pixel with the minimum charge. These 13 figures refer to the SAS PATTERN codes 0, $1-4$, $5-8$, and $9-12$, respectively).

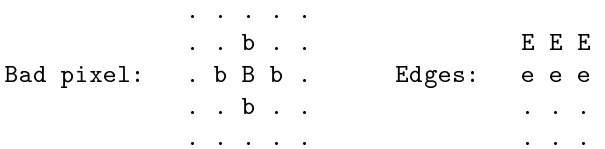
7.4.1.2 Counter examples: Any other configuration of 3 or 4 pixels is most likely caused by pattern pileup (not excluding the possibility that already the 13 patterns mentioned above may suffer from pileup, e.g. if two photons hit the same pixel during one readout frame). These are not considered as recognizable patterns. See also Sects. 7.4.2 and 7.4.5 for further details on the following counter-examples:



	single	double	triple	quadruple
PAT_TYP	1	2	3	4
PAT_IND	0	> 0 for 'X', < 0 for 'x'		
PAT_ORI	0	always > 0		
PI	A	sum(a) for 'X', NULL for 'x'		

After the CTI and gain corrections, in the entry for the corrected amplitude, 'X' will get the sum of the corrected amplitudes of all parts of the pattern, while the corrected amplitude for 'x' will be "NULL".

7.4.1.3 Near bad or edge pixels: A bad pixel is defined as a pixel which is excluded from telemetry; an edge pixel is a 'virtual pixel' just outside the sensitive area, in particular a pixel at the edge of the CCD.



Edge pixels are treated similar to bad pixels. Here 'E' is the area just outside the CCD, and 'e' are the outermost pixels. Since 'E' acts like 'B', 'e' will be treated like 'b'.

Any single event which is located at 'b' will be considered as invalid ('B' is the position of the bad pixel), as will be any recognizable pattern with the maximum amplitude at 'b'. Recognizable patterns with their maximum outside 'b' will be treated as valid events. The reason for this treatment is that for events with their maximum at 'b', part of the charge distribution may be hidden in 'B'. This is less likely, if the maximum is located outside 'b'. Some typical singles and doubles at an edge:

```
                E e . .
                E e . .
single event    E X . .
                E e . .
                E e . .

                E e . .    E e . .    E e . .    E e . .
                E e . .    E x . .    E e . .    E e . .
double pattern  E x X .    E X . .    E X x .    E X . .
                E e . .    E e . .    E e . .    E x . .
                E e . .    E e . .    E e . .    E e . .
```

The first (left) example above for double patterns is considered as valid double, the other three not.

Flags for invalid events / patterns (recognizable case):

	single	double	triple	quadruple
PAT_TYP	-1	-2	-3	-4
PAT_IND	0	always < 0		
PAT_ORI	0	always $= 0$		
PI	-A	-sum(a) for pixel with maximum amplitude		

7.4.2 Unrecognizable event patterns which are no MIPs:

7.4.2.1 Isolated case: Although a unique reconstruction of the number and energy of the photons which have caused such a pattern is not possible, the amplitudes may contain useful information, e.g. if the energy of the photons is known already. Thus, PAT_TYP is positive for all pixels. To distinguish this case from the recognizable pattern case, PAT_IND is always negative. (The main pixel can be found after the column PI has been filled, since this will contain the negative sum of the corrected energies for this pixel, and zero for the others.)

	triple	quadruple	larger pattern
PAT_TYP	3	4	n
PAT_IND	always < 0		
PAT_ORI	always $= 0$		
PI	-sum(a) for pixel with maximum amplitude		

7.4.2.2 Near bad or edge pixels: Assuming that the 'missing area' of the bad pixels which is covered by an unrecognizable event pattern is usually small compared to the size of the pattern (i.e. that only a small amount of the total charge is lost in bad pixels), the treatment is the same as for the isolated case.

7.4.3 MIPs:

If the whole pattern contains at least one pixel with an amplitude above a certain threshold (e.g. 3000 adu), this pattern is interpreted as having been caused by a minimum ionizing particle (MIP), and all related pattern parameters are NULL and `PHA_CTI` and `PI` are NULL, to indicate that these events should be ignored.

7.4.4 Trailing Events:

If an events has been flagged by as trailing event, all related pattern parameters are NULL and `PHA_CTI` and `PI` are NULL, to indicate that these events should be ignored.

All four PAT parameters are combined into a `PAT_ID` (Int16), `PAT_SEQ` (Int8, completely used to store values 1...198) and `PATTERN` (Int8) column which are part of the PPS product 'EPIC pn calibrated event list'.

7.4.5 SAS coding via PATTERN and FLAG:

Any events that have obtained a negative charge sum `PI` (see above) and a `FLAG` > 0 get a positive total charge `PI` assigned; if these have `FLAG` $= 0$ then they get a positive total charge `PI` assigned and the `INVALID_PATTERN` bit set in the FLAG column.

In the implementation of version 6.30.3 (and earlier) single events close to edges or bad pixels do not get PATTERN=0 but get the $8^{\rm th}$ bit set, too, and thus PATTERN=128 (Sect. 7.2). All (apparent) singles at the CCD edges, readout window, or close to bad pixels get PATTERN=128. Additionally, they get the corresponding flag set, e.g. CLOSE_TO_READOUT_WINDOW. So there is a double-coding, the $8^{\rm th}$ PATTERN bit says *that* a FLAG bit is set and the FLAG says *which* one. If one is interested in "safe" single events (as far as the location with respect to CCD edges, readout window, and bad pixels is concerned) then the selection expression "`PATTERN==0`" is already sufficient. This implys that one will not keep single-pixel photons with RAWX=1 or RAWX=64, so there will be larger gaps between the CCDs in such images. If this is not desirable then select "`(PATTERN==0)||(PATTERN==128)`". Then one has to keep in mind the effect of possible charge losses: for broad-band imaging purposes this may be fine, however, accurate spectral modelling should use "`FLAG==0`".

The first example in Sect.7.4.1.3 was considered as valid double (here PATTERN=4), the others not and therefore do not have PATTERN=1-4 but get PATTERN=205: next unused PATTERN, i.e. 13, plus the $7^{\rm th}$ and $8^{\rm th}$ bit set. Similarly, triples (and quadruples) with the main charge at the CCD egde, readout window, or bad pixel, do get PATTERN 14+64+128 = 206 (and 15+64+128 = 207, respectively), and the relevant FLAG setting.

The last invalid quadruple example in Sect.7.4.1 where the mimimum charge is not opposite to the main charge, is an invalid quadruple, so it gets a "base" pattern of 15 plus the $7^{\rm th}$ bit set, i.e. PATTERN=79.

Any very large pattern, with $n \geq 5$ pixels involved, gets a PATTERN = 64+32+(n-5) if no edges or bad pxiels are involved, and PATTERN = 128+64+32+(n-5) if an edge or bad pixel is touched by this event ensemble.

Note: as of version 6.30.4 PATTERN values of 128 have been changed to 0 (i.e. $8^{\rm th}$ bit is not set anymore for singles), and PATTERN values of 205 have been changed to 1–4 (i.e. $7^{\rm th}$ and $8^{\rm th}$ bit are not set anymore for doubles). All the FLAG coding and all other PATTERN bit coding is unchanged.

Note: a keyword `SCR_88`="T" in the `EXPOSURE` (and `EVENTS`) extension of the output file of **epevents**

indicates that this new behaviour was used, if the keyword is absent or contains "F" then the old behaviour (PATTERN=128 or PATTERN=205 for singles and doubles close to CCD edges or bad pxiels) is present.

8 Algorithm

```
subroutine epevents

*** Get program parameters from parameter input

*** Open and check input file, copy to output file and condition outfile
infile   = dataSet(stringParameter("eventset"),READ)
  check if infile has EVENTS extension
  check if infile has EXPOSURE extension
copyDataSet infile => outfile
outfile = dataSet(stringParameter("outset"),MODIFY)
  create event attributes in EVENTS extension
  addColumn DETX,DETY,PHA\_GAIN,PHA\_CTI,PI,PAT\_TYP,PAT\_IND,PAT\_ORI,
              PAT\_ID,PAT\_SEQ,PATTERN
  setNullValue PHA\_CTI,PI,PAT\_TYP,PAT\_IND,PAT\_ORI,PAT\_ID,
              PAT\_SEQ,PATTERN
  open columns needed
  do checks on quadrant and ccd numbers

*** Get program parameters from dataset attributes
  get datamode to guide further processing
     EpnPrimeFullWindow
     FastTiming          => no trailing event rejection
                            source position used to determine event time
     FastBurst           => source position used to determine event time
  get number of events in data set
  get window dimensions and determine edges
  get source position for timing and burst mode
  get MIP rejection mode used in epframes
     NONE = mip columns not flagged
     SAS  = mip columns flagged
     COM  = mip columns and neighbours flagged

*** Read and convolve bad pixel map
  check if bad pixel extension exists
  create bad pixel map (intensity codes bad pixel type)
  create convolved bad pixel map (mark pixels above, below, left and right
    of a bad pixel, and mark edge pixels)

*** Determine frame boundaries and discarded lines in frame
  read number of frames and frame boundaries from exposure extension
  read discarded lines in frame from exposure extension

do frame = firstFrame, lastFrame

   set parameters and allocate array size according to number of events
       in frame

   *** adjust bad pixel map and convolved bad pixel map according to
```

```
      mip rejection mode (MIP flagged columns correspond to bad pixels,
      neighbour columns are flagged like edge pixels)

    *** flag reemission events (if not timing mode)
      determine trailing event
      flag if PHA of trailing event is < reemissionthreshold
        determined from PHA of preceding event
      count trailing events

    *** perform pattern analysis
      determine the pattern parameters PAT_TYP, PAT_IND, PAT_ORI
        and PAT_SEQ
      mark uncertain patterns as invalid: if a single or the maximum of a
        pattern coincides with a pixel of the convolved bad pixel map,
        edge pixels are treated as if there was a bad pixel just outside
        the sensitive area of the CCD
      check for MIPs
      count different pattern types

    *** collect PAT_Infos and combine into PATTERN and PAT_ID columns

    *** perform gain correction, use CAL routine CAL_pnGainCorrect

  enddo

  *** perform cti correction, use CAL routine CAL_pnCtiCorrect

  *** recombine the corrected amplitudes from each pattern (=> PI column)

  show results of trailing event and pattern analysis

  *** convert raw unlinearized detection coordinates RAWX/Y into
  corresponding physical location DETX/Y

  *** Store statistics of pattern analysis as attributes in exposure extension

  end subroutine epevents
```

9 Comments

- The TIMING mode of the EPIC pn needs a different way of split event recombination. This has still to be investigated.

10 Future developments

As the pattern recognition is a time consuming part (however, much less than the CTI correction) of **epevents**, the user might not want to repeat this if e.g. the calibration changed. If the parameter `patternanalysis` is equal 'no', the **epevents** task will be restarted only after the pattern analysis. The parameter is not active yet.

The rejection of MIP events can be done as part of the on-board event processing. This results in a rejection of columns where such events have been detected. The discarded lines information (from onboard CR detections) is the sum over several (default 20) frames. For full correction **epevents**, however, would need this information per CCD/quadrant and frame for its pattern analysis. Therefore depending on on-board preprocessing the pattern products from **epevents** can have higher or lower quality.

References

[1] K. Dennerl, U.G. Briel, F. Haberl, G. Hartner, N. Krause, M. Popp, and V.E. Zavlin. Determination and correction of the charge transfer efficiency of the pn-ccd camera. In O.H.W. Siegmund and K.A. Flanagan, editors, *EUV, X-Ray, and Gamma-Ray Instrumentation for Astronomy X*, volume 3765 of *Proceedings of SPIE*, pages 232–243. SPIE, 1999.

[2] SSC. XMM Survey Science Centre to Science Operations ICD for SSC Products. Technical Report XMM-SOC-ICD-0006-SSC Issue 2.1, SSC, Mar 2000.

[3] Giuseppe Vacanti. Interface specification for the xmm event attributes, April 1999. GV/XMM/1999-01.