emenergy

August 16, 2005

Abstract

Assign energy and quality flag to events for one node of one EPIC-MOS CCD over one exposure.

1 Instruments/Modes

Instrument	Mode
EPIC MOS	IMAGING, TIMING

2 Use

pipeline processing	yes
interactive analysis	yes

3 Description

emenergy is the task which prepares the spectral analysis of EPIC-MOS data. It adds a quality flag to all events, prepares the (optical) background map, computes the best `PHA` energy from all information available, corrects for charge transfer losses and translates it into `PI`. The CCDBKG subroutine cannot be applied to slew data.

emenergy calls (in order) the following subroutines, all of which can be individually switched off:

- EVFLAG flags all events whose pattern could not be fully recognized on-board because of the proximity of:
  1- a border of the CCD (`CLOSE_TO_CCD_BORDER`)
  2- the left or lower border of the current window (`CLOSE_TO_CCD_WINDOW`). In window mode, the right and upper borders are analysed including the lines/columns next to them.
  3- a pixel declared as bad in the on-board library (`CLOSE_TO_ONBOARD_BADPIX`)
  4- a line (`CLOSE_TO_BADROW`) or column (`CLOSE_TO_BADCOL`) deactivated through a large offset (voluntarily or not)
  5- a dead pixel/line/column (`CLOSE_TO_DEADPIX`)
  6- a bright pixel not declared in the on-board library (`CLOSE_TO_BRIGHTPIX`)

Only bad pixels immediately next to the main pixel of the event (not diagonal) justify flagging. If the CCD is read through both nodes (known from the window keywords) then the limit between nodes must be flagged as well (CLOSE_TO_NODE_BOUNDARY).
In case 5 (missing data) of dead pixels at distance 2, then PERIPIX should be incremented for the CCDBKG task. Single events in that case would not be flagged at all. Similarly in case 4 (wrong data) at distance 2, then PERIPIX should be incremented by 8 (to deactivate the event for CCDBKG).
In TIMING mode there are no bad pixels, only bad columns. In Compressed Timing mode, events for which RAWX and PATTERN were not known are flagged as UNKNOWN_RAWX_PATTERN.
The flag value allows to identify the reason for flagging (using binary coding as described in **evatt**, explicited in the XMMEA_nn keywords). The flags set by EVFLAG are informative (contrary to those set in **emevents** which are mostly rejection flags). The flagged events should be used or not depending on the type of study. For spectral applications aiming at the best resolution and most reliable energies it is safer to reject them. The effective area should then reflect that. For detection purposes they can be included to improve the statistics.
Finally, the events outside the field of view are flagged (OUT_OF_FOV) to allow easily removing them from sky maps. They are useful to estimate the particle background, though.

- CCDBKG prepares the CCD background map used by MAKE_E to correct the energies. By default (no outside information required) a map of the CCD background can be formed by averaging E_4/(16-PERIPIX) for all events encompassing a pixel in their 5x5 area. Only unflagged events should be considered. Large events (> 2 pixels) are not suitable because the charge leakage into E_4 is usually non-negligible. Events at distance 1 or less from the borders include under/overscans in the outer corona and should not be used. Events with PERIPIX > maxperipix are not used either. The default (maxperipix=0, no other event in the outer corona) is rather strict to ensure the best quality, and may be relaxed in the case of a piled-up source to maxperipix=2. Events with E_1 < minenergy are also not used. This is because a fraction of low energy events are noise events with a wrong value of E_4 (rejected by REJECT_E3E4 down the line).
The map is built in RAWX/RAWY coordinates. To check for possible time variations CCDBKG outputs E_4 averaged over the CCD (or window) as a function of time.
If fillccdbkg=yes then the off-source areas of the CCD (where less than fillminnumber good events contribute) are replaced by the sum of averages over the current line and current column (off-source), minus the full average. This is equivalent to replacing the integer offsets onboard by real (higher precision) ones. This filling procedure is applied separately to the areas of the CCD within the field of view and outside the field of view.
To make the background determination more robust, REJECT_E3E4 is called a first time (with no background map) before CCDBKG to remove the most obvious noise events, and a 3 sigma clipping is applied to E_4 locally (with respect to a first estimate of the background). This procedure does not apply in REDUCED IMAGING mode (no E_4 information), nor in TIMING mode.

  In parallel (useccfdarkframe parameter) one can use the background map from the average dark frame in the CCF (of better spatial resolution). All the E_i are then corrected for the difference between the dark map and the local offset, before building the internal background map (as described above) and building the PHA energy (in MAKE_E). This allows to correct CCD defects at the pixel scale (whereas E_4 is an average over 16 pixels around the X-ray hit).
Another option is to use a specific background map instead if data in DIAGNOSTIC mode have been recorded prior to the scientific exposures (not PPS). The decimal digits of the reference background (in ADU) are subtracted from PHA and PI correctly, but not from the E_i (integers). Consequently if the PHA column is regenerated later from the same events list it will lose the decimal precision.

- REJECT_E3E4 (IMAGING mode, not reduced IMAGING). Some instrumental effects result in wrong events associated with pathological charge values next to it. This information is

known through the E_3 and E_4 data. The REJECT_E3 procedure in **emevents** rejected part of that noise. An additional way to reject it is to reject all events whose E_3/E_1 and E_4/E_1 (after CCD background subtraction) lie outside the normal cloud, specified as an ellipse in that plane (the ellipse parameters are in the E3E4ELLIPSE_n fields of the `XMM_MISCDATA` CCF file). This is done with flag `BAD_E3E4`, and has little effect on true X-rays (10% loss at 100 eV, $< 1\%$ above 150 eV).
This test cannot be applied directly next to rows or columns with a large offset (taken from the `OFFSETS` extension). In that case E_3 is first corrected for the large offset. E_4 can be corrected as well if the offset is not too large (such that E_4 was truncated at -16384) and `PERIPIX`=0 (otherwise one cannot know how many pixels of the bad row/column were counted in E_4). If E_4 cannot be corrected it is set to 0 for the test (*i.e.* the test bears only on E_3/E_1).

- MAKE_E computes a single energy `PHA` (in ADU) for each event from a weighted sum of the E_i, and the residual background $Bkg(x, y)$ computed in CCDBKG, assumed not to vary with time (*i.e.* the time series output from CCDBKG is not used). This is performed via CAL_mosPhaBuild, described in **calmosalgo**. If large offsets (from the `OFFSETS` extension) exist next to the event, E_1 and E_2 are corrected for that before being sent to CAL_mosPhaBuild.
  Events with reconstructed `PHA` above the upper EMDH threshold are flagged as `OUTSIDE_THRESHOLDS`.
  MAKE_E adds `TLMIN/TLMAX` keywords to the `PHA` column. By default, randomisation over 1 ADU is performed on E_1 before building PHA. Randomisation may be switched off using `randomizeenergy`.
  In TIMING mode MAKE_E does nothing but the randomisation (the sum was already performed in flight).

- CTI_CORR. Charge transfer losses will inevitably appear at some level over time due to radiation damage. They occur as the lines are transferred, first to the frame store area, then to the reading register, and as the pixels are transferred along the reading register. Knowledge will undoubtedly accumulate with time on this phenomenon. In theory, charge losses could be local, could depend on energy and on count rate. This is performed via CAL_mosCtiCorrect. The result is written into `PI`, so that `PHA` always remains uncorrected for CTI.
  In Timing mode `RAWY` does not correspond to the vertical position on the CCD, but to time. The CTI correction should be called assuming the source position, given by the `SRC_RAWY` keyword. In Compressed Timing mode, the source position should be taken from the `SRC_RAWX` keyword when `RAWX` is unknown (events flagged as `UNKNOWN_RAWX_PATTERN`).

- ENERGY. The energy of each event in ADU units (from MAKE_E or CTI_CORR) is converted into pulse invariant (`PI`) units. This is performed via CAL_mosGainCorrect. Events with reconstructed `PI` above the upper EMDH threshold (converted into PI) are flagged as `OUTSIDE_THRESHOLDS`. ENERGY adds `TLMIN/TLMAX` keywords to the `PI` column.

4 Parameters

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

Parameter	Mand	Type	Default	Constraints

ineventset	yes	dataset	' '	none

name of input events file

newoutput	no	boolean	no	yes/no

Create output events file (no for overwriting input)

outeventset	no	dataset	'events.out'	none

name of output events file. If this parameter is set, then `newoutput`=Y is automatic

flagevents	no	boolean	yes	yes/no

activate EVFLAG

correctcti	no	boolean	yes	yes/no

activate CTI_CORR

getccdbkg	no	boolean	yes	yes/no

activate CCDBKG

minenergy	no	integer	100	≥ 0

Minimum PHA energy (in ADU) for CCDBKG

maxperipix	no	integer	0	$\geq 0, \leq 7$

Maximum `PERIPIX` for CCDBKG

fillccdbkg	no	boolean	yes	yes/no

Fill the empty areas of CCD background map

useccfdarkframe	no	boolean	no	yes/no

Use dark frame in CCF

fillminnumber	no	integer	10	≥ 0

Number of events below which filling is applied

maskedccdset	no	dataset	' '	none

name of CCD masked image file from emdiag (for CCD background)

rejectbade3e4	no	boolean	yes	yes/no

Reject events with bad E3/E4

makepha	no	boolean	yes	yes/no

activate MAKE_E

randomizeenergy	no	boolean	yes	yes/no

randomize PHA over 1 ADU

correctgain	no	boolean	yes	yes/no

activate ENERGY

writebackgroundset	no	boolean	no	yes/no

Create output file for optical background image and time series

backgroundset	no	dataset	'bkgccd.map'	none

name of output file for optical background image and time series

timebin	no	real	100	> 0

time bin for E4 time-series

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.

getparamvalues01 *(error)*
: blank input event file name

getparamvalues03 *(error)*
: PUT_XY not run before

getparamvalues06 *(error)*
: MAKE_E not run and `correctgain`=Y

readBkg01 *(error)*
: unexpected data type in diagnostic background map

readBkg02 *(error)*
: array too small in diagnostic background map

readBkg03 *(error)*
: not 2-D array in diagnostic background map

readBkg04 *(error)*
: diagnostic background map not compatible with event file

readBkg05 *(error)*
: `useccfdarkframe`=yes and missing offset/variance extension

evflagImage13 *(warning)*
: `flagevents`=yes and a bad pixel has invalid coordinates in IMAGING mode. It will be ignored
corrective action: inform SOC (this is probably an error in the CCF)

evflagTime13 *(warning)*
: `flagevents`=yes and a bad pixel has invalid coordinates in TIMING mode. It will be ignored
corrective action: inform SOC (this is probably an error in the CCF)

subDark10 *(warning)*
: background subtraction from the CCF or an external file modifies the energies. It cannot be run twice on the same data
corrective action: restart from the output of **emevents** if you want the full precision of the background subtraction

readBkg12 *(warning)*
: dimensions of the diagnostic background map larger than CCD size. Use anyway
corrective action: check this is not a mistake in the input image file

getparamvalues10 *(warning)*
: no `EXPOSURE` extension. The CCD-specific keywords will be forgotten after merging the CCDs
corrective action: restart with **emevents** if you want the `EXPOSURE` extension

getparamvalues11 *(warning)*
`flagevents`=yes and EVFLAG already done before. Not repeated
corrective action: restart from the output of **emevents** if you really want EVFLAG to run anew

getparamvalues12 *(warning)*
`useccfdarkframe`=Y or `maskedccdset` set in Timing mode. Not done
corrective action: these options cannot be used in Timing mode

getparamvalues13 *(warning)*
Reduced Imaging data and `getccdbkg`=Y. Not done
corrective action: this option cannot be used on Reduced Imaging data

getparamvalues19 *(warning)*
`makepha`=Y and background subtraction from the CCF or an external file already done before. Decimal digits will be lost
corrective action: restart from the output of **emevents** if you want the full precision of the background subtraction

getparamvalues20 *(warning)*
the event file is empty. Nothing done
corrective action: check this is normal

getparamvalues22 *(warning)*
not Imaging data and `rejectbade3e4`=Y. Not done
corrective action: this option can be used only in full Imaging mode

getparamvalues26 *(warning)*
slew data and `getccdbkg`=Y. Not done
corrective action: this option cannot be used on slew data

6 Input Files

1. event file (from **emevents**) with bad pixels (`BADPIX`), offset/variance (`OFFSETS0`), bad offsets (`OFFSETS`) and `EXPOSURE` extension.
2. optical background map from Diagnostic mode in **emdiag** format (interactive only)

7 Output Files

1. event file with modified/additional items:
   - filled in columns `FLAG`, `PHA` and `PI` (for PRODUCT: EPIC event list)
   - modified columns `ENERGYE1`, `ENERGYE2`, `ENERGYE3` and `ENERGYE4` if `useccfdarkframe`=Y
   - `XMMEA_nn` keywords detailing which flags were set
   - `XMMEA_SM` keyword defining selection for spectra
   - keywords detailing which subroutines were activated
   - comment lines with names of input files
   - keywords propagated to the `EXPOSURE` extension (for **evlistcomb**)
2. optical background map and time-series over the CCD from E_4 (for calibration purposes) as a separate FITS file with:

- the background map (real, 600x600) with the offsets added back in, stored in the primary.
- an array (integer, 600x600) extension (CCD_WGHT) storing the weight (number of pixels contributing) of each pixel
- a binary table extension (CCD_VAR) for the time-series, with two columns BKGCHARGE (real: average background per time bin, offsets subtracted) and BKGWEIGHT (integer: number of pixels contributing to each time bin).
- an OFFSETS0 extension propagated from the events file, with the same structure as the offsets file in the ODF ([1]).

8 Algorithm

```
   subroutine emenergy

   Read the file names among parameters

   Opening of input file event
   Copy of input file event on output file event if need be

   parameters read : choice of sub_tasks to execute
   Loop over input task parameters
      Read the task parameter / 1 to perform the procedure,
                                0 not to perform it  /
   End loop

   Get the values of files columns in memory

   call EVFLAG

   call REJECT_E3E4 with no background
   call CCDBKG
   call CLIPE4
   call CCDBKG
   discard flags applied by intermediate calls to REJECT_E3E4 and CLIPE4

   call REJECT_E3E4
   call MAKE_E
   call CTI_CORR
   call ENERGY

   Close files

 SUBROUTINE EVFLAG
   if evflag requested  then
      Loop over events
         flag events using binary coding for different reasons
      end loop
   endif

SUBROUTINE CCDBKG
   if useccfdarkframe then read dark frame map
   else if maskedccdset is set then read DIAGNOSTIC map
```

```
   Subtract the offsets
   Correct the Ei columns by the contents of the reference map
      integrated over the area associated to each Ei
   if  ccdbkg requested then
      Loop over events
        except for flagged events, events with peripix > 0, pattern > 4,
                   E1 < minenergy
        calculation of sigmae4 = summation of energy E4
        calculation of sigmaper = summation of (16-peripix)
      end loop
      divide sigmae4 by sigmaper to obtain BKG value
      if fillccdbkg then
        Loop over positions where less than fillminnumber events contribute
           Average E4 over rows rowav and columns colav
           Replace BKG by rowav + colav - full average
        end loop
      endif
      if bkgfile NE ' ' then
         save of the arrays sigmae4 and sigmaper in a new fits file
      keywords are recopied
   endif

SUBROUTINE CLIPE4
    Get variance on E4 - BKG(x,y) after applying same selection as in CCDBKG
    Flag events where E4 is more than 3 sigma away from BKG(x,y)

SUBROUTINE REJECT_E3E4
    if reject_e3e4 requested then
      Read e3cen,e4cen,e3rad,e4rad,e3ang from CAL
      Loop over events rows
         loce3 = E3
         loce4 = E4
         if bad offsets nearby then
            loce3 = loce3 + projection of additional offsets onto pixels in E3
            offe4 = sum of additional offsets in outer corona
            if peripix==0 and offe4 < 16000  loce4 = loce4 + offe4
         endif
         loce3 = (loce3 - BKG * (8- number of pixels in E2)) / (E1-BKG) - e3cen
         loce4 = (loce4 - BKG * (16-PERIPIX)) / (E1-BKG) - e4cen
         dist2 = (loce3*cos(e3ang)+loce4*sin(e3ang))/e3rad)**2 +
                 (loce4*cos(e3ang)-loce3*sin(e3ang))/e4rad)**2
         Flag events with dist2 > 1
      end loop
    endif

SUBROUTINE MAKE_E
   if make_e requested
      call CAL_integerToReal(ener1,e1rand)
      edum = E2
      if bad offsets nearby then
         e1rand = e1rand + additional offsets on central pixel
         edum = edum + projection of additional offsets onto pixels in E2
      endif
      call CAL_mosPhaBuild(e1rand,edum,pattern,pha,BKG,E3,E4,peripix,flag)
   endif
```

```
SUBROUTINE CTI_CORR
   if cti_corr requested  call CAL_mosCTIcorrection

SUBROUTINE ENERGY
   if  energy requested  call CAL_mosGainCorrect

   end subroutine  emenergy
```

9 Comments

- The flag map built by EVFLAG could be of interest to **eexpmap** and **arfgen**. To that end it would need to be supplemented with small 5x5 maps (one per pattern) describing the effective extension of a bad pixel. Computing the area lost to bright pixels declared on-board is another matter. It depends on energy, through the bright pixel level.
- If the telescope's attitude were to vary significantly (more than the PSF) with time, then it should be considered using X,Y (attitude corrected) as coordinates for CCDBKG, since the optical background from stars would follow the attitude variations. Note that this would need accumulating the map over a larger area than the current window to accommodate pointing excursions. The large scale optical background (*e.g.* stray light from the earth) does not come through the telescope's optics and therefore has no reason to follow the attitude variations. On the other hand it would be expected to vary along the orbit. The way to deal with that would be to model the large scales (e.g. by a quadratic shape) separately in several time bins and then model the parameters' variations with time (quadratic).
- The subroutines CTI_CORR and ENERGY only call the CAL routines CAL_mosCTIcorrection and CAL_gainCorrection.
- The scheme does not allow for the possibility of measuring the gain in one observation (with the calibration source) and passing this on to **emenergy** (to supplement or replace the CAL).
- The baseline is to perform the ENERGY subroutine at the end. It could also be considered performing ENERGY (converting all E_i into `PI`) before CTI_CORR. This would be the best option if the ADU to `PI` conversion were significantly non linear. On the other hand it is somewhat more difficult to calibrate and may induce some additional error.

10 Future developments

- It should be considered scanning the E_4 time-series to warn the user if significant variations are detected.

References

[1] ESA. XMM Interface Control Document: Observation and Slew Data Files (XSCS to SSC) (SciSIM to SOCSIM). Technical Report XMM-SOC-ICD-0004-SSD Issue 2.5, ESA/SSD, June 2000. Found at the URL: `ftp://astro.estec.esa.nl/pub/XMM/documents/odf_icd.ps.gz`.