eexpmap

April 16, 2023

Abstract

Create EPIC exposure maps to be used by the tasks **emask**, **esplinemap**, **eboxdetect**, **emldetect**, **ewavelet**, and **esensmap**.

1 Instruments/Modes

Instrument	Mode
EPIC MOS:	IMAGING
EPIC PN:	IMAGING

2 Use

pipeline processing	yes
interactive analysis	yes

3 Description

Using CCF data on the spatial quantum efficiency, filter transmission, mirror vignetting, and field of view, instrument maps containing the spatial efficiency of the instrument are constructed. Quantum efficiency, filter transmission, and vignetting are evaluated assuming an event energy which corresponds to the mean of the PI channel boundaries specified by the command line parameters `pimin` and `pimax` (*note, that, depending on the source spectrum, this may introduce errors if very wide [pimin,pimax] intervals are used; create narrow band exposure maps instead and weight appropriately*). Alternatively, the `PI` channel boundaries will be read directly from the data subspace extension of the input image (not yet implemented). The inclusion of the telescope vignetting in the exposure calculation can be switched off, if `withvignetting` is set to 'false'. Bad pixels as listed in the bad pixel extension of the input file are excluded from the instrument maps. EPIC PN offset columns (as specified in the *offsets* extension of the photon event list) are set to zero in the exposure map, if they had been removed from the input reference EPIC image. Depending on flag selections in the image, the surroundings of bad pixels and border pixels are also excluded from the instrument maps.

From version 4.0 on, the parameter `badclean` is removed from the parameter list, and the pixels in the neighbourhood of bad pixels, CCD borders, and offset columns are treated according to the flag selections of the input image. The flag selections are read from the DSS keywords of the input image.

From the instrument maps, exposure maps are constructed which may be either output in detector or in sky coordinates (this is the default; see parameter `withdetcoords`). Note that the input image has to be of the same coordinate type (detector/sky) as the required output image. In the case of sky coordinates, the attitude file generated by the task **atthkgen** is rebinned. A new attitude bin is started when the change in attitude exceeds the required positional accuracy (parameter `attrebin`). The integration time is calculated from the good time intervals valid for each chip and is corrected for subsequent time selections performed by the user. The exposure falling in each time bin is obtained from the exposure extension of the input dataset.

An attitude histogram is created from the rebinned attitude file, and for each attitude bin, the corresponding exposure values are finally projected onto the sky and 'accumulated' into the respective sky image pixels. The resulting exposure maps will thus contain the exposure (in units of seconds) for a particular EPIC sky image.

The following filters are read from the data subspace keywords of the input images and are taken into account for the calculation of the exposure maps: `TIME` range filters, `GTI` filters, `CCDNR` range filters, `FLAG` range filters, `FLAG` bit-mask filters. The task tries to determine the exposure time by looking for `GTI` extensions in the input image of the form `STDGTInn`, `STDGTIn`, `STDGT`, or `GTI`. If no `GTI` extension is found, the exposure time is taken from the `EXPOSURE` keyword. No merging of multiple gti is performed in the usedss=false case but the first set of `GTI` extensions in the sequence above is used. Note that the task expects the gti information in an extension of the input **image** whereas the `BADPIX` and `EXPOSURE` extensions are read from the input event file. The `usedss`=false setting is mainly useful for the processing of non-SAS-derived input datasets. **In the case of SAS-derived images it is strongly advised that the parameter `writedss` of task evselect is set to true**. For EPIC MOS input images, the effective frame time is taken from the column `TIMEDEL` of the `EXPOSURE` extensions in order to correct for mode dependent dead-times. In the case of EPIC PN, the keyword `TIMEDEL` is used. Since the value of `TIMEDEL` incorporates mode dependent corrections for out-of-time events, a keyword OOTCORR=true is written to EPN exposure maps in order to avoid double correction by **emldetect**. The keyword `OOTFRAC` contains the ratio between the keywords `TIMEDEL/FRAMETIM`.

Task **eexpmap** supports the calculation of several exposure maps in different energy bands in one run of the task. The exposure maps are used in the EPIC detection chain by the tasks **emask**, **esplinemap**, **eboxdetect**, **emldetect**, **ewavelet**, and **esensmap**.

3.1 Matching exposure maps and event coordinates

Earlier versions of **eexpmap** often produce exposure maps with zero exposure at certain image pixels where the event count is nonzero in the input image. This is especially so when the input image is produced from an EPIC event list with randomized coordinates (the default), and for sky images. This usually appears as an offset between the input image and the exposure map, but in no particular direction, and no shifting between the two can make them match exactly. For exposure maps in detector coordinates DETX/Y, the maximum 'offset' is ± 1 image pixels (default: ± 4 arcsec). For sky maps, the maximum offset depends in part on the attitude information specific to each observation, but is usually less than ± 2 image pixels.

Bugs related to this problem have mostly been fixed since **eexpmap** 4.6.1 of xmmsas 9.0.1. **eexpmap** should now produce exposure maps in DETX/Y that match input images exactly. The single exception is when the event list from which the input image is made contains any PN event with RAWX=28 and DETX=5476, or with RAWY=72 and DETY=9588. The DETX/Y values in these cases actually lie outside the RAWX/Y pixels. Removing these events, reassigning their DETX/Y, or turning off randomization will all solve the problem.

To produce exposure maps in sky coordinates that match input images exactly, users should call **eexpmap** with the parameter `attrebin` set to at most 10^{-7} radian, i.e., `attrebin=0.020626481` (arcsec) or smaller. For long observations, this could prolong the running time of **eexpmap** substantially. However, exposure maps thus produced will match the input images exactly. Doing the same with older versions of **eexpmap** will also help to reduce, but will not completely eliminate this offset.

4 Parameters

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

Parameter	Mand	Type	Default	Constraints

imageset	yes	filename	image.fits	

Name of EPIC FITS image

attitudeset	yes	filename	attitude.fits	

Name of attitude file

eventset	yes	filename	events.fits	

Event file, providing bad pixel and exposure extensions

expimageset	yes	list of file-names	expimage.fits	

Name(s) of output exposure image(s)

withdetcoords	no	boolean	false	

If true, the exposure map will be output in detector coordinates. In this case, the input image(s) have to be binned in detector coordinates DETX, DETY.

withvignetting	no	boolean	true	

If true, the exposure map will include vignetting

usefastpixelization	no	boolean	true	

If true, a speed increase of up to a factor of two is achieved, at the cost of inaccurate exposure values in border pixels

attrebin	no	float	4.0	[0.0<param<60.0]

Positional accuracy of attitude rebinning in arcseconds. Changes in the attitude less than `attrebin` are ignored when rebinning the attitude data. Set `attrebin`=0.020626481 (i.e., 10^{-7} rad) or smaller to ensure that the output sky exposure map matches event lists.

pimin	no	integer	2000	[0<param<30000]

Lower `PI` energy boundaries of exposure images

pimax	no	integer	4500	[0<param<30000]

Upper `PI` energy boundaries of exposure images

usedlimap	no	boolean	false	

If true, use discarded line maps provided by **epexposure**

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.

MissingParameter *(error)*
Missing input file name

FileMismatch *(error)*
Inconsistent number of input images

FileMismatch *(error)*
Inconsistent instruments or bands

WrongInst *(error)*
Unknown instrument

badDatamode *(error)*
IMAGING mode data required for this task

NoGTI/noExposure *(error)*
No `GTI` extension and no `EXPOSURE` keyword found

ArrayOutOfRange *(error)*
DSS contains more than 5000 time intervals

noGTI *(error)*
No `GTI` or `TIME` filter in DSS

noGTI *(warning)*
No `GTI` extension found in input image
corrective action: Look for `EXPOSURE` keyword; assume one GT interval of duration given in `EXPOSURE`

NumGTI *(warning)*
Number of `GTI` extensions /= number of chips
corrective action: Assume same `GTI` for all chips

NoBadPix *(warning)*
`BadPixel` extension not found
corrective action: Create exposure map without bad pixels

NoExpoExt *(warning)*
Exposure extension not found
corrective action: Assume 100 % exposure in each `GTI`

NoFilt *(warning)*
No `FILTER` attribute found
corrective action: Assume open filter position

NoSubMode *(warning)*
No valid `SUBMODE` attribute found
corrective action: Assume full window mode

undefinedHelpVector *(warning)*
Help vector has undefined length
corrective action: Taking aspect solution as help point to continue

MissingAttribute *(warning)*
Keyword is missing in input file
corrective action: Keyword is not copied to output file

NullValues *(warning)*
NULL values in the attitude table were ignored
corrective action: NULL values will be ignored

NoOffset *(warning)*
No EPN offset extension found
corrective action: no offset treatment done

6 Input Files

1. PPS product (from task **evselect**): EPIC FITS image (Instrument ID, Mode/Submode, filter ID, GTI, WCS keywords; reading of other DSS filters not yet implemented)
2. from task **atthkgen**: Attitude file
3. event file (`EXPOSURE` and `BADPIX` extensions)

7 Output Files

1. PPS product (to be used by tasks **emask**, **esplinemap**, **eboxdetect**, **emldetect**, **ewavelet**, **esensmap**): EPIC exposure images (one per energy band)

8 Algorithm

```
LOOP over attitude file from task \task{atthkgen}

  Rebin attitude according to positional accuracy requirement
  specified by parameter attrebin.

  LOOP over chips

    For each chip, merge rebinned attitude bins with GTI
        and with time selections performed on the data

    Get exposure in each time bin from EXPOSURE extension

  END loop

END loop
```

```
Creation of instrument map and exposure map

LOOP over CCDs

  LOOP over detector pixels (PIXCOORD1)

    Check if pixel is a bad pixel (from BADPIX extension), border
    pixel, or outside FOV (CAL_getFOVmap)


    Depending on flag selection keywords in the input image,
    check if pixel is neighbour of bad pixel or border pixel

    EXIT loop if one of the above is true

    Obtain quantum efficiency for each energy band
    (CAL_getQuantumEfficiency)

    Transform to PIXCOORD2

    Obtain filter transmission (CAL_getFilterTransmission)
    and vignetting (CAL_get EffectiveArea) for each energy band

    For each energy band, multiply quantum efficency, filter
    transmission and vignetting and write to instrument map

    LOOP over attitude histogram

      Project detector pixels onto sky

      Multiply instrument map with exposure in attitude bin
      and distribute into sky pixels

    END of attitude loop

  END of detector pixel loop

END of loop over chips

Write exposure map to output
```

9 Comments

10 Future developments

- DSS support: Currently only `TIME` range and `GTI` filters as well as `CCDNR` range filters are evaluated. DSS filtering of other event properties (spatial, pattern, energy ...) still needs to be implemented.
- Coordinate transformations: Starting with version 4.6.1, **eexpmap** repeats the same errors as are

in **attcalc**, and converts with different formulae for MOS and for PN, in order to match exactly the event coordinates calculated by **emevents** and **epevents**. These discrepancies and errors await resolving.

References