eexpmap

August 15, 2005

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'. The event pattern types for which the quantum efficiency is calculated may be read from input parameter `pattern` and alternatively (not yet implemented in this version) directly from the data subspace of the input dataset. 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 the parameter `useoffset` is set to 'true'. Depending on the value of parameter `badclean` the surroundings of bad pixels and border pixels are also excluded from the instrument maps. The surrounding to be omitted is given by the following grid (pixels with values $\leq$ badclean value are omitted):

```
5  4  3  4  5
4  2  1  2  4
3  1  0  1  2
4  2  1  2  4
5  4  3  4  5
```

From version 3.29 the parameter `badclean` has default value 0 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 the the input image has be of the same coordinate type (detector/sky) as the required output image. In the case of sky coordinates, the attitude file generated by 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 (if enabled by parameter `usedss`) 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. If parameter `usedss` is set to false, 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 eventfile. 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 parameter `usedss` and the corrsponding parameter `writedss` of task evselect are 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 dependend dead-times. In the case of EPIC PN, the keyword TIMEDEL is used. Since the value of TIMEDEL incorporates mode dependend 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**.

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 (for input of bad pixel and exposure extensions)

expimageset	yes	list of file-names	expimage.fits	

Names of output exposure images

withdetcoords	no	boolean	false	

If true the exposure map will be output in detector coordinates. In this case, bin type of input images has to be DETX, DETY.

usedss	no	boolean	true	

Set to true to make the program read time selections performed on the dataset[2]

withvignetting	no	boolean	true	

if set to true exposure map will include vignetting

usefastpixelization	no	boolean	true	

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

withpatterns	no	boolean	true	

If true event patterns will be read from parameter 'pattern'; if false the data subspace extension of the dataset will be accessed instead (not yet implemented)

pattern	no	string	' '	

Event pattern types may be specified in the following format (example): '0 1 2' for pattern types 0 to 2 (0 referring to single events) [1]

badclean	no	integer	0	[0<param<5]

Cleaning around bad pixels and borders: Value 1 will exclude 4 immediately neighbouring pixels; value 2 will also exclude diagonal pixels (see section description)

attrebin	no	float	4.0	[0.0<param<60.0]

Positional accuracy parameter (arcseconds), changes in the attitude less than `attrebin` are ignored when rebinning the attitude data.

useoffset	no	boolean	true	

use information on EPN offset columns

hducut	no	float	0.0	[0.0<param<1000.0]

threshold for column offset; columns with offsets > hducut will be excluded

pimin	no	integer	2000	[0<param<30000]

Lower energy boundaries of exposure images

pimax	no	integer	4500	[0<param<30000]

Higher energy boundaries of exposure images

[1] Note that in the current calibration database PN pattern type 1 refers to all non-single events.
[2] DSS support is currently limited to TIME range and gti filters, as well as to CCDNR selections. **Note that parameter `writedss` of task evselect must be set to true when creating the input image for task eexpmap**.

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 DSS 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 bad (from BADPIX extension), border pixel,
    or outside FOV (CAL\_getFOVmap)


    Depending on parameter badclean or flag selections keywords in 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; add contribution for each
    selected pattern type)

    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.

References