epileupmask

January 12, 2017

Abstract

This task makes a mask image of piled-up pixels.

1 Instruments/Modes

Instrument	Mode
EPIC	Imaging

2 Use

pipeline processing	yes
interactive analysis	yes

3 Description

3.1 What is pile-up?

Pile-up is in a way a simple phenomenon. However once the image/event is obtained, it is not trivial at all to reproduce the non-piled up image, which is what an observer really wants.

Pile-up is essentially described as an event which contains more than one incoming photon in it, primarily due to slow read-out, comparing with the incoming flux. By definition, those more-than-one photons are inseparatable. Therefore practically the event is regarded as the '*sum*' of those events; *ie.*, the energy of the event is regarded as the total sum of those incoming photons. Therefore these piled-up events (1) skew the spectrum of the source, namely, the piled-up spectrum appears to be harder than the original spectrum, and (2) reduces the observed flux in photons per unit time for the entire energy band (though preserves in the first-order approximation the flux in unit of energy per unit time). These two are the most simple effect of the pile-up.

In the case of the CCD detection (in the imaging mode), the situation is more complicated. If the point-spread function (PSF) is larger than the CCD pixel, which is the case in both pn and MOSs, the second (third, fourth, etc) photon, which is due to cause the pile-up, is more likely to fall onto the surrounding pixel than the central pixel, onto which the first photon fell. If the first photon is due to

cause a single pixel event, the combination of the first and second photons would cause a double pixel event — either diagonal pixel or bipixel event, or, depending on the geometry and energy, it may cause tripixel or quadripixel event. Or worse, if the pile-up is extremely severe, the sum of all those piled-up photons may give an active signal for the entire 3×3 pixels (pattern 31 in MOS), which leads to the situation where the position is likely to appear to be a void spot in the image, because the pattern 31 is generally filtered out in the filtering.

Hence the pile-up also alters the grade-branching ratio. In short higher grade (more double-pixel events, for example) events are more likely to increase. This follows if one extracts only the single pixel events, the derived flux will be smaller than the true value in the case of pile-up, because more events are detected as double (triple, quadruple) pixel events as piled-up (or combined) events, each of which could be a sum of originally single-pixel events.

In the piled-up image, the observed PSF must be flatter than the non-piled-up one, because the centre of the PSF is more heavily piled up than its outer skirts. If the pile-up at the centre is very severe, as explained above, the centre of the PSF may show a depression, or worse a void.

What we get as data is the already piled-up data, that is, a skewed image with a skewed energy spectrum (and light curve). Generally speaking we know neither the true image nor spectrum. Conversely that is what we want to get. We understand well the skewing process by pile-up, however the inverse function for it is not analytical and is extremely hard to calculate even computationally.

Consequently the pile-up is a function of the incoming flux, energy, surface brightness (of the source(s)), as well as the shape of the PSF. It affects the grade-branching ratio (of the CCD), observed flux and energy and spatial distribution. If the source flux is the input, the vignetting is a factor of concern, too.

3.2 What does this task do to handle pile-ups?

The purpose of this tool is to give a mask file, where those pixels that are likely to be affected with pile-up are marked, based on the user-specified threshold, applied to the normalised count-per-frame-per-pixel map, after calculating the averaged grade-branching ratio.

To calculate the normalised count-per-frame-per-pixel map, this task, if possible, only takes into account the single-pixel events. In the calculation it weighs each event, depending on the energy, normalising to that in Al K line at 1.487 keV, where 77.5% and 67.8% of all events at this energy should be observed as a single pixel in MOSs and pn, respectively, if no pile-up occurs. For example, an event at the energy, where the single-pixel branching ratio is 38.8% (= 77.5/2), is weighted as twice as much as those at 1.487 keV.

3.3 What is the limitation of this task, if any?

First, this task works on the basis that **pile-up rarely happens**. If only a count rate in a pixel is given, even if the count rate is quite small, such as 0.001 cts/s, it is impossible to tell whether the pixel is piled up or not — it is either hardly piled up or very severely piled up (*e.g.*, Ballet 1999). This task assumes always the former case, *i.e.*, the count rate is small. In other words, the heavily piled-up pixels could be regarded as not being piled up. If there is a very bright and hence heavily piled-up point-like source, the centre of the source position could be regarded as non-piled-up, whereas an annular region centred at the source will be marked as piled-up.

Second, one does not know whether an event of interest is a piled-up or clean one. The task calculates on the basis that all the events are clean ones. For example, an event with the energy of 3-keV is calculated as such, whether it is caused by a real 3-keV photon or actually a combination of a 1-keV and a 2-keV

photons.

Third, this task only takes into account an averaged count rate in the given duration, where the duration is either the whole observation period or that during which the attitude of the satellite is stable, depending on the command-line arguments. Even when a source becomes bright enough to cause a significant amount of pile-up at one stage (*i.e.*, outburst) during a period, if the duration of the outburst phase is short enough, comparing with the whole period, the averaged count rate, which is the total count divided by the total exposure (of the period), could be below the threshold. This may happen not only for variable sources but also due to the moduration of the spacecraft attitude (again depending on the command-line arguments).

Fourth, the event file given should not be filtered based on the energy. Events in the whole energy band is essential to estimate piled-up pixels. Bad GTI events is allowed to be filtered, providing the given exposure extention/attiribute reflects the GTI information. Grade selection can also be made, if a user wishes — remember this task uses only the single-pixel events. The background events should be included in the event file, as long as it is a single-pixel event[1].

3.3.1 Behaviours and command-line arguments.

This task accepts either an event or image file as an input (controlled by `inputstyle`), and outputs either the sky coordinate image or 3-dimensional chipcube (controlled by `outputstyle`).

In short, we recommend to specify an event file as an input, and the sky coordinate image as an output, of which the reasons are described below.

1. if the input is an event file and the output is the chipcube, the process runs as follows:
   The task calculates the normalised count-per-frame-per-pixel (=rate) map for each chip, taking account of the energy (PI) of each event. Then the pixels, where the rate exceeds the given threshold, are masked and it is output.

2. if the input is an event file and the output is the sky-coordinate image, the process runs as follows:
   The core of the routine is exactly the same as the above case. However, the coordinate conversion into sky-coordinates is not a trivial job, as the pile-up phenomenon is basically processed on the chip(raw) coordinates. Hence a user is allowed to specify the binned-attitude file. If this (`binnedattset`) is set, the task splits the event files into the given time bins (described in the file `binnedattset`), performs the above-described calculation (of the normalised rate map) for each of these time bins, converts each normalised rate map and mask map into the sky coordinates[2]. Finally the task calculates the logical sum of the mask file. In other words, all the pixels that experience the rate above the threshold in one of the time bins are masked. Note that frames with too short duration are not used in this calculation of rate and mask images (Controlled by `avcnttoprocess`) in this case. If a user specifies so (controlled by `attstyle`), the entire observation is used as one set — in that case, the spatial fluctuation of the spacecraft is not taken into account (the median attitude value is simply used for the coordinate conversion).

3. if the input is an image file and the output is the chipcube,
   the process is similar to the case 1. However the energy of each event is unknown. Therefore the image is assumed to have a single energy, which is the mean energy of those specified by the user (Controlled by `evlo` and `evhi`). As for the attitude of the spacecraft, which is

[1] Of course, there is no way to distinguish the origin of any particular event between the source and background!
[2] By definition, the attitude of the spacecraft during each time-bin is stable *enough*. Therefore the produced sky coordinate image should be accurate *enough*.

used in converting the sky coordinates to the chip ones, the median attitude value is simply used.

A potential problem is when the image bin size is larger than the chip-coordinate one, the output mask on the chip-coordinates will be sparse, because only the central value of the image coordinates are taken into account in the current algorithm. For example, suppose the former is twice as larger than the latter, and all the image pixels exceed the piled-up threshold, then only one in every 4 pixels in the chipcube is flagged as piled-up.

4. if the input is an image file and the output is the sky-coordinate image, too,
the task does the same thing as the above case, although the output file is in the same coordinates. However, the spacecraft attitude information is still needed to get the (number of) frame information for each pixel. The median attitude value is used for that purpose.

Note the algorithm is identical to that in **eootepileupmask**. Consult its manual for detail.

4 Parameters

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

Parameter	Mand	Type	Default	Constraints
ratethreshold	yes	real	0.008	0 <=`ratethreshold`
The threshold of the rate, where the pixels having the rate larger than this value are regarded as 'bad'.				
inputstyle	no	string	evlist	evlist—image
Whether the input file is eventlist or image.				
eventset	no	dataset		
Name of the input event list dataset used to construct the mask image. This parameter is read if `inputstyle`='evlist'.				
imageset	no	dataset		
Name of the input image dataset used to construct the mask image. This parameter is read if `inputstyle`='image'.				
evlo	no	real		0 <`evlo`
The lower energy bound of `imageset`. This is mandatory only if `inputstyle`='image'.				
evhi	no	real		0 <`evhi`
The upper energy bound of `imageset`. This is mandatory only if `inputstyle`='image'.				
issingleeventonly	no	boolean	yes	yes—no
This is read if `inputstyle`='image'. If the image includes only the single pixel events, this must be true. Otherwise (false), all the other standard grade events (*e.g.*, 0–12 in MOSs) are assumed to be included in the image.				
outputstyle	no	string	sky	sky—raw
If 'sky', the OOTE map is output in sky coordinates, to the file referred to by parameter `ooteimageset`. In this case a template set (`templateset`) is needed and the `attstyle` parameter is also read. If `outputstyle`='raw' on the other hand the output is written to a cube (in the `expcubeset` format) to the file pointed to by `ootecubeset`.				

templateset	yes	dataset		

This parameter is read if `outputstyle`='sky'. This file should contain an image in the primary extension, which is used to define the pixel dimensions and World Coordinates of the output image.

masksset	no	dataset	pileupmask.ds	

If `outputstyle`='sky', the output mask image in sky coordinates is written to this file name. If `outputstyle`='raw', the output mask cube image in chip coordinates is written to this file name.

attstyle	no	string	binnedset	binnedset—template

This parameter is read if `outputstyle`='sky'. To convert from chip to sky coordinates it is necessary to know the spacecraft attitude. However the attitude is never completely stable and may vary significantly during an exposure. In this case the nett sky image must be a mosaic of components from different values of the attitude. A time series of attitude values (such as that made either by **attbin** or **evproject**) can be supplied to parameter `binnedattset` if `attstyle` is set to 'binnedset'. If it is judged that the attitude wander during the exposure did not exceed some small fraction of the image pixel dimensions, or if the binned attitude set is not available, then the user may choose to set `attstyle` to 'template' instead. In this case a single fixed value of attitude is read from *_PNT keywords in the template image header.

binnedattset	yes	dataset		

If `attstyle`='binnedset' the user should supply to the present parameter the name of a dataset which contains a time series of the spacecraft attitude variation during the exposure. The user should be aware of the fact that if an attitude (time) bin (specified in this file) is too small, the time bin is unlikely to be processed (see the description of `avcnttoprocess` below). Therefore it is recommended to create the `binnedattset`, allowing a relatively large attitude fluctuation, such as 1 arcsec, so that attitude bins are not too finely split.

avcnttoprocess	no	real	7.0	0 <=`ratethreshold`

When `attstyle`='binnedset', the supplied `binnedattset` file may provide time durations with very short exposures. In that case if a pixel happens to have an event during the short duration, the value of count/frame at that pixel in that duration is large and likely exceeds the threshold `ratethreshold`, and as a result that pixel will be masked as piled-up just because the pixel has one event at a wrong time. To avoid this happening, any time duration with too short exposure should not be processed. This parameter gives the threshold for it, *i.e.*, only when there are reasonable number of frames in the time duration, in which the averaged count, corresponding to the piled-up threshold, exceeds this parameter `avcnttoprocess`, the time duration is processed to calculate the output mask. For example, if `ratethreshold`=0.008 and `avcnttoprocess`=7.0, each time duration should have the (maximum) number of frames larger than 875 so as to be used by the process, which corresponds to ∼2.3 ks for MOSs. Note that because the single-event ratio at 12 keV in MOSs is ∼0.23 (0.54 for pn), `avcnttoprocess`=7.0 corresponds to just above 2 count per frame per pixel in MOSs for very high energy photons, after the grade-branching ratio is corrected.

withrateimage	no	boolean	no	yes—no

If this is true, the rate imageset calculated is also output with the filename of `ratesetroot`. If `outputstyle`='sky', the rate image set is in sky coordinates, whereas if `outputstyle`='raw', the rate set is the cube format in chip coordinates. Note that the rate is corrected for the event at the energy of 1.487 keV.

ratesetroot	no	dataset	rateset	

This is read only if `withrateimage`=true. See the section of `withrateimage` for detail.

withboresightfudge	no	boolean	yes	yes—no

Flip the sign of the boresight euler%psi. **This parameter will be removed** after the boresight is fixed.

selexprstyle	no	string	userranges	userranges—dss—userexpr

Use of task **epileupmask** implies that the user wishes to model the background component of a real image. To do this properly it is necessary that the OOTE map and the image reflect the same selection

of events. It is therefore necessary to provide details of the event selections used to construct the real image. Ideally the user should supply these in the form of the Data Subspace (DSS) of the actual image by selecting `selexprstyle`='dss' and then supplying the file name of the image with the DSS to parameter `dssset`. However it has been found convenient to also allow the user to supply the event selection expression directly (via `expression`) or simply to choose to supply a set of energy ranges. The latter can be done by selecting `selexprstyle`='userranges' and then supplying lists of values to `evlo` and `evhi`. Note that in this circumstance the assumption is made that no other significant non-spatial selections were made to create the original image.

dssset	yes	dataset		

I `selexprstyle`='dss', information about event selections is sought in a Data Subspace (DSS) of the primary extension of this dataset.

expression	yes	string		

This parameter is read if `selexprstyle`='userexpr'. It should contain the selection expression used to construct the original image.

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.

dummy *(error)*
DUMMY

6 Input Files

1. (Mandatory) a dataset with an exposure cube (without vignetting) in the primary image extension. The output of task **eexpchipmap** is suitable. A description of the cube format can be found in the documentation of that task.
2. (Only mandatory if `inputstyle`='evlist') A calibrated event list for the relevant EPIC camera, created by either **emchain** or **epchain** or alike.
3. (Only mandatory if `outputstyle`='sky') a FITS dataset, which contains an image in its primary extension. The name of this dataset should be supplied to parameter `templateset`. The output image (`noiseimageset`) is constructed so as to match `templateset`'s pixel dimensions and World Coordinates.
4. (Only mandatory if `outputstyle`='sky' and `attstyle`='binnedset') a file output by **attbin**, containing a table `ATT_BINS` with columns `TSTOP`, `RA`, `DEC`, `PA` and `IS_GOOD`. The table should also contain a `TIMEZERO` keyword.

7 Output Files

- If `outputstyle`='sky':
  1. `ooteimageset`: a 2-byte-real-valued OOTE map, in sky coordinates, is contained in the primary image extension.

  This dataset contains the same keywords in the primary HDU as the template image, except for DSS-related keywords. Extra extensions in the template image are not propagated.
- If `outputstyle`='raw':
  1. `ootecubeset`: an OOTE-map cube is contained in the primary image extension.

  The format of this cube is described in the task documentation of **eexpchipmap**.

8 Algorithm

**********Not yet written.

9 Comments

References