arfgen

February 1, 2016

Abstract

Creates an OGIP-compliant Ancillary Response File (ARF) for the source in question.

Contents

1 Instruments/Modes

Instrument	Mode
EPIC MOS	IMAGING, TIMING
EPIC PN	IMAGING, TIMING

2 Use

pipeline processing	no
interactive analysis	yes

3 Usage and Examples

Further information about the scientific use and accuracy of this task may be found in the document, XMM-SOC-PS-TN-0043, (available from *http://xmm.vilspa.esa.es/external/xmm_sw_cal/calib/documentation.shtml#EPIC/*)
In order to execute **arfgen** correctly, the user should provide the following:

1. The appropriate values for the environment variables `SAS_CCF` and optionally `SAS_ODF`.
2. An input spectrum generated by **xmmselect** or **evselect** with the parameter `writedss` = true,
3. a source position (optional). The user can use either the target position available from the input spectrum dataset by setting `withsourcepos` to `false`, or specify a different one by setting `withsourcepos=true`, and entering the coordinates and coordinate system through the parameters `sourcex`,`sourcey` and `sourcecoords`. (section 5.4)
4. If the detector map (a grid of positions used to calculate and flux weight the ARF: section 5.3) is either `psf` or `flat`, the map bounds and pixel size may be specified. For these types of map, the map bounds and pixel size by default will be automatically set to cover the regions over which the spectrum is accumulated, although the user has the option to set them explicitly if desired via the `withdetbounds` , `withdetbins` and associated parameters. Note that if a `psf` detector map is used, the energy at which the PSF model is taken may be specified. The user can supply an external detector map instead (section 5.3.2), but the bounds must cover the regions.

The user must then decide which models or corrections to apply (section 5.1), whether to use the energy grid from the CAL (`withrmfset=false`), or from an input RMF (`withrmfset=true`) [section 5.6] Finally, the user can, optionally, choose the name of the output ARF dataset through the `arfset` parameter.

The following sub-sections show examples of usage.

3.1 Example 1: Point source

Here is a typical example of running **arfgen** based on a spectrum of a point source:

```
arfgen spectrumset=spectrum.ds detmaptype=psf
```

Generates a dataset `ancillaryfile.ds`. The effective area, filter transmission and quantum efficiency curve are included in the ARF.

A point source located at the centre of the source region described in the data subspace within *spectrum.ds* is assumed. An encircled energy correction is applied (section 5.5). Corrections are applied for any CCD gaps or source region area falling outside the field-of-view or observing window but not for bad pixels or bad columns.

The ARF is emission-weighted using an internal detector map based on the PSF. The map is rectangular-shaped, and has a weighting which falls off with distance from the source in line with the PSF.

3.2 Example 2: Correcting for bad pixels

To correct for bad pixels and bad columns (section 5.10) a file containing the bad pixel extensions must be specified on the command line by:

```
arfgen spectrumset=spectrum.ds badpixlocation=myevents.ds
```

The bad pixel information is, at least initially, stored in the event files.

By default the code looks for pixels which fall outside the CCD collecting area and outside the field-of-view with a resolution of 2 arcseconds if the `extendedsource` parameter is set and 0.5 arcseconds for a point source. In some cases, for example if the majority of the source region lies outside the field of view, or a significant fraction of the counts from a point source fall within a chip gap, then this resolution is too coarse. It may be changed by:

```
arfgen spectrumset=spectrum.ds badpixlocation=myevents.ds badpixelresolution=1.0
```

which improves the accuracy of the bad pixel calculation for an extended source. The downside is that an increased resolution will cause the execution time of the code to increase substantially.

3.3 Example 3: Point source with explicit position

Here is another example with an explicit source position – (1000,-1000) in DET coordinates – specified:

```
arfgen spectrumset=spectrum.pha
withsourcepos=true sourcecoords=det sourcex=1000 sourcey=-1000
```

See also 5.4 for details of why the task needs the source position, and how to change it.

3.4 Example 4: Specifying an external detector map for an extended source

Here is an example illustrating how to pass an externally-created detector map (section 5.3.2) to the task:

```
arfgen spectrumset=spectrum.pha detmaptype=dataset detmaparray=detmap.ds
extendedsource=yes
```

Generates a dataset `ancillaryfile.ds`, weighting the arf spatially using the first array found in the dataset `detmap.ds`. The effective area, filter transmission and quantum efficiency curve are included in the ARF. An ARF for an extended source is generated.

Such detector maps can be generated in **xmmselect** given an event list that contains a column DETX, DETY, by selecting those columns in the main GUI dialog of that task, clicking the 'Image' button and then specifying the preferred image binning in the subsequent **evselect** dialog.

Alternatively, it could be generated by invoking **evselect**:

```
evselect table='0001_0000010010_M1S00100IME.FIT:EVENTS'
destruct=false withfilteredset=true
withimageset=true imageset=detmap.ds xcolumn=DETX ycolumn=DETY
withxranges=true ximagemin=-1500 ximagemax=1500 withyranges=true
yimagemin=-1500 yimagemax=1500 imagebinning='imageSize'
ximagesize=20 yimagesize=20
writedss=true updateexposure=true
```

3.5 Example 5: Using an RMF dataset to specify the energy grid

By default an ARF is generated at a set of energies specified within the CCF and is compatible with the default energies produced by rmfgen. To use different energies, contained within an rmf file, m1_r5_all_15.rmf, use the command:

```
arfgen spectrumset=spectrum.pha detmaptype=dataset detmaparray='detmap.ds'
withrmfset=true rmfset=m1_r5_all_15.rmf
```

3.6 Example 6: Setting the BACKSCAL value

```
arfgen spectrumset=spectrum.pha setbackscale=yes
```

This command calculates the area of the source region and writes it into the BACKSCAL keyword in the spectrum header. This may also be achieved by calling the metatask **backscale**.

3.7 Example 7: Getting the unvignetted ARF

To return the unvignetted ARF the source position needs to be forced to be the position of the optical axis. This can be achieved by defining it in telescope coordinates.

```
arfgen spectrumset=spectrum.ds withsourcepos=yes sourcecoords="tel" sourcex=0
sourcey=0 filterdss=no withdetbounds=yes withbadpixcorr=N modelee=N
detmaptype=flat detxbins=1 detybins=1
```

This returns the on-axis effective area with no PSF or bad-pixel correction.

3.8 Example 8: Finding the cross-region ARF

```
arfgen crossregionarf=yes spectrumset=spec_output.ds crossreg_spectrumset=spec_input.ds detmaptype=datas
detmaparray=image200.ds
```

This command returns the effective area in a region of the detector (defined in spec_output.ds) coming

from the PSF overflow of flux from the sky region defined in spec_input.ds. The detmaparray, image200.ds, should be an image which includes both detector regions with a pixel size which is small compared to the size of the regions. It is recommended that the number of detmaparray pixels which falls within the spec_output.ds region is > 300 to properly sample the region. But please note that this calculation is very compute intensive and very fine detector maps will lead to a very long run time.

4 Definitions

For the sake of brevity, the following terms will be used in this document:

- 'EQPOS' refers to equatorial coordinates [ra, dec].
- 'DET' coordinates refers to the CAMCOORD2 system, but expressed in units of *0.05 arcsec* instead of mm, as these are the units in which the detector coordinates are presented in the event list after pipeline processing (ie columns `DETX` and `DETY`; see [5]).
- 'POS' coordinates refers to the projected sky coordinates of the source relative to the nominal pointing position, in units of 0.05 arcsec. This is the system in which the sky coordinates of an event are presented in the event list after pipeline processing (ie columns `X` and `Y`; see [5]).
- 'TEL' refers to telescope coordinates (theta, phi).

5 Description

Generates an OGIP-compliant ([4]) ancillary response file (ARF). The ARF contains a table listing area values (the column `SPECRESP`) for a number of different energy ranges (the columns `ENERG_LO` and `ENERG_HI`). The ARF can be used in conjunction with the response matrix file (RMF, generated by **rmfgen**) to allow fitting of particular spectral models against the observed spectral data. The actual choice of fitting package is entirely up to the user, the only prerequisite is that it is OGIP-compliant. The most widely used package is expected to be Xspec.

Specifically, **arfgen** takes calibration information provided by the CAL, performs the necessary corrections for instrumental factors depending on the user's requirements, and the state of spectral data, and formats the output such that it is compliant with [4]. It is this operation that is described here; details of individual CCF constituents and how the CAL processes such data are beyond the scope of this document, and the reader should refer to the Calibration Access and Data Handbook for such information ([1]).

The following table summarises the factors used to generate the ARF, and the status of the corresponding code needed to model that factor in **arfgen**:

Item	Description	Status
1	Telescope Effective Area	Implemented
2	Filter Transmission	Implemented
3	Quantum Efficiency	Implemented
4	Complex region selection	Implemented
5	Spatial variation of source	Implemented
6	Source position specification	Implemented
7	Encircled energy correction (point source)	Implemented
8	RMF dataset i/p optional	Implemented
9	Pattern support	Implemented
10	Flux loss due to pile-up	Not implemented
11	Bad pixel correction	Implemented
12	Bad column correction	Implemented
13	CCD gaps	Implemented
14	Out of FOV/observing window	Implemented
15	Out-of-time event smearing (PN)	Implemented

Item 10 will be considered in future versions and is discussed in section **??**.

Whether a factor is taken into account by **arfgen** in the generation of the ARF can be controlled by individual parameters with the prefix `model`. For example, to correct for filter transmission, the parameter `modelfiltertrans` should be set to `true`.

5.1 Effective Area, Filter Transmission, Quantum Efficiency and Grating attenuation

arfgen simply constructs the appropriate quantities from the CAL, in the form of vectors of discrete energy-dependent elements, and combines these data multiplicatively to form the ancilliary response. The energy grid used to construct these vectors is taken either from the CAL, or from the input RMF dataset (see section 5.6).

Note for MOS instruments, a factor ~ 0.5 of photons are redistributed away from the detector by the RGA. This is already taken into account by the CAL in the call to CAL_effectiveArea.

The user can enable or disable the effective area, filter transmission, QE or grating attenuation by setting the relevant parameter (eg `modeleffarea`) to `true` or `false` respectively.

Note that the values in the effective area curve are dependent on the source position with respect to the focal plane. See section 5.4.

5.2 Spatial region settings

The spectrum is usually accumulated by the user over a specific region or regions in space. In order for the ARF generated to be appropriate for that region or regions, **arfgen** extracts spatial information from the Data Sub-Space (DSS) stored in the spectral dataset (see the **dsslib** documentation for more information regarding the DSS). This task uses the regions to calculate the spatially dependent elements of the ARF.

The DSS information is only written to the spectrum by **evselect** (or **xmmselect**) during its creation, if the **evselect** parameter `writedss` is set to `true`, and if the user has specified a region expression via the **evselect** `expression` parameter that is anything other than `true` or null. If there is no DSS information

present an ARF will not be calculated.

Note that **arfgen** only handles regions, for imaging mode data, defined in POS (X/Y) or DET (DETX/DETY) space; so specifying a region component in terms of any other axis or axes, eg RAWX/RAWY, will cause the task to fail. Timing and Burst mode regions, on the contrary may be specified in terms of RAW coordinates.

5.2.1 Using an image mask

A spectrum may be produced by filtering events using a 2-dimensional mask in either sky or detector coordinates. This is supported for the calculation of the ARF of an extended source. To maintain the resolution of the image mask it is important to use a finely binned detector map within **arfgen** (see section 5.3). Preferable is a detector pixel image which just encompasses the masked region, otherwise a flat detector map will be ok.

e.g.

```
arfgen spectrumset=spectrum.ds extendedsource=yes detxbins=400 detybins=400
detxoffset=10000 detyoffset=10000 withdetbounds=yes
```

The chip gap and bad pixel/column corrections will be correctly calculated if the detector map is fine enough. The **backscale** task will also produce the correct area calculation in this case.

5.3 Spatial variation of source

5.3.1 A detector map

Given that some of the ARF components vary as a function of position, the spatial variation of a source should be considered in the generation of an ARF. This information can be passed to **arfgen** by specifying a dataset containing an image of the source distribution in DET coordinates (a so-called 'detector map'). **arfgen** would then compute a weighted ARF based on the variation of the ARF for each pixel in the detector map, using the values in the detector map as a weighting. Ideally this map should represent a model of the incident source distribution, *convolved* with the telescope PSF, but with no other instrument effects applied (it fact it would be more preferable if the input source distribution did not include the PSF convolution either; but PSF convolution is unfortunately beyond the scope of **arfgen**). However, in most cases it is expected that the *observed* source imprint will suffice.

The detector map also specifies the spatial points at which **arfgen** extracts ARF information from the CAL. These points correspond to the centre of each pixel.

NB: The detector map image must be created in detector coordinates in the current implementation. It is a priority to allow this file to be created in sky coordinates in a future release.

5.3.2 Different types of detector map

A detector map is used to provide a grid of positions over which to calculate and flux weight the ARF. The default detector map is 'flat' which has a weighting of 1.0 for each detector map pixel and is rectangular having sides which just encompass the region specified in the DSS. Only detector map pixels falling within the source region are included in the ARF calculation, e.g. for a circular source box, the pixels in the

corners of the detector map will be ignored. The number of detector map pixels is set automatically unless explicitely set with the parameters `detxbins` and `detybins`. The calculated number of pixels in each dimension is constrained to be between 5 and 80, and is set proportionately to the *total* extent of the extraction region. The total extent is taken from the included *and* excluded regions and for example can be surprisingly large if detected sources have been removed by excluding regions.

Some fine regions, e.g. a narrow annulus at a large off-axis angle, may not be well represented by the default map. In these cases more map pixels can be introduced by specifying e.g. detxbins=100, detybins=100, but beware that the task execution time increases with the number of bins.

arfgen supports 3 types of detector map, selectable through the `detmaptype` parameter:

- `flat`:

  This is the default internal representation where all pixel values are set to 1.

  Given that all the pixels are of the same value, the function of the flat detector map is essentially to describe the spatial grid points which the task should consider during the computation of the ARF.

  The map bounds are by default matched to the extent of the selected regions, but can be set explicitly by setting `withdetbounds` = `true` and then choosing values for the parameters `detxoffset`, `detyoffset`. The map is centred on the source position. The automatically calculated number of pixels can be overridden using the parameters `detxbins` and `detybins`. This map should be used for extended sources unless a detector map image is supplied.

- `psf`:

  This is an internal map based on a model of the PSF taken from the CAL. This is the most accurate for point sources where the majority of the flux falls at the centre of the source region. In practice it is unlikely to produce a very different ARF to the 'flat' map, except at large off-axis angles where the calibration parameters can change rapidly.

  The extent and coarseness of the detector map may be specified on the command line similarly to that of the 'flat' map.

  Given that the PSF is energy dependent, the energy at which the PSF model is taken may also be specified by the user, through the parameter `psfenergy`.

- `dataset`:

  This is a user-defined detector map. This allows the user to have control over the spatial distribution of the source which is particularly useful for extended sources.

  This must be in the format of an image array of a dataset, and that image array must contain WCS information that describes the mapping onto DET coordinates. Such maps can be generated in **xmmselect** given an event list that contains a column `DETX`, `DETY`, by selecting those columns in the main GUI dialog of that task, clicking the 'Image' button and then specifying the preferred image binning in the subsequent **evselect** dialog. Alternatively, it could be generated directly from **evselect** - section 3.4 provides an example of doing this.

  A detector map of this form can be passed to the task through the `detmaparray` parameter.

  Note that in order for **arfgen** to generate an output consistent with the data, the detector map must enclose the regions over which the spectrum is accumulated; one or more warnings are raised if the extent of the selected regions exceeds the bounds of the detector map along either the X and/or Y axes.

  NB: The detector map may not be specified in X/Y coordinates in the current release.

5.3.3 Detector map filtering

In practice, **arfgen** does not use the input detector map 'as is'; instead it uses a filtered form of the detector map in which the points that lie outside the regions used to accumulate the input spectrum are excluded.

A consequence of this is that the user must ensure that the source regions lie within the bounds of the input detector map, otherwise the output ARF could be grossly inaccurate. The task performs a crude check to see if regions are compatible with the detector map, and issues a warning if this is not the case. The user is welcome to be conservative and create a detector map over a very large window, enclosing the regions plus a lot more. In this case the number of image map pixels should be correspondingly greater to ensure a good coverage over the region of interest, which will result in a performance overhead during the initial filtering process.

The filtered map is available as a by-product, by setting `withfilteredset` = `true` and specifying a name for the output dataset via the parameter `filteredset`. The map is provided in pixel list format. The filtering operation can be switched off, if desired, by setting `filterdss` = `false`. In this case, the regional information is ignored, and *all* pixels in the input detector map are written to the 'filtered map', which in turn means that all pixels are considered during the ARF computation.

arfgen supports region information specified in terms of DET or POS (sky) coordinates. In the former case, the region specification can be used to filter the DET-based detector map directly. In the latter case, an additional step is necessary to transform the region specifications to the DET system. The transformation is performed by making a call to the task **attcalc**.

5.3.4 Performance considerations

The performance of **arfgen** depends mainly on the number of points in the detector map that are enclosed within the said regions. This means that the coarser the detector map, the less CAL calls are made, and the less accurate the resultant ARF is. At the other extreme, a detector map that is too finely binned would result in a large number of CAL calls and would affect the execution time of the task. Ideally the binning should be such that it reflects the spatial scaling of instrumental variations across the detector, including any chip gaps and bad columns which fall within the extraction region.

The execution time is also dependent upon the actual size of the source region if chip gap and bad pixel corrections are switched on (`withbadpixcorr`=true). In fact this becomes the dominant contribution for large source areas.

5.4 Source Position

This task needs as input, in addition to the spectrum and detector map, the position of the source in question. This is currently used to determine the effective area (5.1), to calculate the encircled energy correction (5.5), to compute bounds for the flat and psf-type internal detector maps, and to construct the psf detector map distribution.

The task gets this information from one of two sources: from the centre of the source region in the input spectrum (if the parameter `withsourcepos`=`false`), or explicitly from the user (`withsourcepos`=`true`).

If the latter, the user can choose to specify the position in one of four coordinate systems: EQPOS, POS, TEL and DET. The coordinate system is specified through the parameter `sourcecoords`. The actual coordinates of the source can be input though the parameters `sourcex` and `sourcey`. These parameters

are essential for complex regions, such as an included circle with several excluded circles, as in these cases the task can not work out the real source centre unaided.

5.4.1 Conversion to DET coordinates

The internal coordinate system used by **arfgen** is DET, so if the input coordinate system is EQPOS, POS or TEL, a conversion to DET is performed. If the input system is POS, it is firstly converted to EQPOS. To perform this intermediate conversion, the nominal pointing position is required. This is taken from the global attributes `REFXCRVL` and `REFYCRVL` of the spectrum dataset.

The conversion from EQPOS to DET in general requires knowledge of the spacecraft attitude. By default the task uses the mean pointing position contained within the spectrum header in the **RA_PNT, DEC_PNT, PA_PNT** keywords. This is expected to be accurate enough in most cases. An alternative approach is to use the attitude from the OAL corresponding to the start of the observation by specifying `useodfatt`=true. In this case, as the OAL is used by **attcalc** and for the coordinate conversion, the environment variable `SAS_ODF`, or the task parameter `odf` need to be set to the appropriate ODF directory in order for the conversion to be correct.

It is recommended to use DET coordinates if the mean attitude over the exposure period differs significantly from the start attitude, say by more than 30 arcseconds.

An intermediate stage of the conversion from equatorial to detector coordinates is via the telescope position (theta, phi). The source position may be input directly in terms of theta, phi by setting `sourcecoords`=TEL, `sourcex`=theta and `sourcey`=phi. In this case, `sourcex`=0 `sourcey`=0 refers to the optical axis.

5.5 Encircled energy correction

In the case of a point source, **arfgen** can correct for the flux that is outside the region(s) over which the spectrum was accumulated. These regions are read from the spectrum's DSS information. The correction works by taking a model of the PSF from the CAL, centred at the source position and calculating the total fraction of that PSF within the regions of interest.

To enable the encircled energy correction for point sources, the parameter `extendedsource` must be set to `false`, as well as `modelee` being set to `true`.

The encircled energy correction is energy dependent, so ideally the EE should be computed for each and every point on the energy grid. However, given that the EE calculation is computationally intensive, the option to resample the energy grid is provided by the parameter `eegridfactor`. The resultant EE correction vector is interpolated back onto the original energy grid, and each ARF element is corrected using the corresponding EE correction factor. See section 10 for further information.

By default **arfgen** uses a fomulation of the PSF as a function of energy and off-axis angle, known as the ELLBETA model, to calculate the encircled energy correction. This model is valid for all extraction region shapes as it works by summing up contributions to the PSF over a grid of pixels which fall within the region. The resolution of this grid (and of the bad pixel calculation) is set by the parameter `badpixelresolution` which has a default of 0.5 arcseconds. The PSF model may be overridden using the `psfmodel` parameter.

5.6 Optional RMF dataset as input

arfgen computes a vector of areas for a given set of energy ranges. These energy ranges are by default taken from the CAL.

Alternatively, the **arfgen** can take the energy ranges from the `ENERG_LO` and `ENERG_HI` columns of an RMF dataset. This feature is useful if, for example, the user wants to make sure his/her RMF and ARF datasets are in sync. This can be done by setting the parameter `withrmfset` to `true` and specifying a dataset name for the `rmfset` parameter.

5.7 Pattern support

arfgen reads in the pattern range from the DSS information in the spectrum dataset, and multiplies the quantum efficiency curve by the fraction of the events which fall in these patterns as a function of energy. If any problem is found with the PATTERN definition in the DSS; e.g. it is missing completely, it is unbounded or it is non-standard then the *entire* range of allowed patterns (0-31 for the MOS and 0-255 for the PN) are assumed. In these cases the warning NonStandardPatterns is issued. The standard ranges are singles (pattern 0 only), doubles (patterns 1-4 only), singles plus doubles (patterns 0-4) and for the MOS singles+doubles+triples+ quadruples (patterns 0-12). All other combinations are considered to be non-standard.

5.8 Timing and Burst modes

These modes are treated as a special case within **arfgen**. For the EPIC-pn camera, regardless of the spatial region used to extract the spectra the source is assumed to be located at a DETY position calculated from the SRCPOS keyword in the spectrum header (or RAWY=190 if this keyword isn't present). The vignetting and encircled energy corrections are applied assuming a point source at this position. Bad pixels, columns and chip gaps are corrected for in the same way as for imaging mode observations.

In Timing mode it is recommended to take the full range of RAW-Y when extracting an EPIC-pn spectrum (in MOS the RAW-Y information is not present). It can be useful to take a restricted RAW-Y range in Burst mode to exclude contamination from the on-axis source. In this case **arfgen** scales the ARF proportionately to the range selected. Such a scaling is not applied for Timing mode data.

If pile-up is suspected in a Timing mode observation then it can be convenient to exclude the inner columns, containing the highest count rate. In this case a selection such as

```
(RAWX in [30:34] || RAW in [37:41])
```

should be used. Here, the inner two columns, RAWX=35 and 36, have been excluded. The encircled energy calculation will correct for the missing columns.

5.9 Flux loss due to pile-up

It is possible to greatly reduce the effect of pile-up by extracting data from an annulus about the source centre. The optimum inner radius to use depends on the strength of the pile-up but a typical first guess would be 20 arcseconds. The task calculates the encircled energy correction for annuli using the accurate, energy-dependent PSF described in section 5.5. The task **epatplot** can be used to investigate pile-up

issues.

While a basic pile-up correction implementation is available, this feature is not currently being maintained.

arfgen provides a feature for correcting for the loss in detection efficiency due to over-lapping events occuring within the same frame. It does not, however, correct for spectral distortion as a consequence of pile-up. This is caused when two events of the same pattern overlap exactly; in this situation, the onboard electronics will recognise the pair as a single event with a charge equal to the sum of the charges due to the individual events. The approach of [2] was used to correct for flux loss. He provides formulae for flux loss for each of the four patterns accepted by the on-board detector electronics, given the pattern morphology and contributions of each pattern to the incident photon spectrum. In order to correct for pile-up, the task requires a rough estimate for the incident photon spectrum. This should be provided by the user in the form of a DAL dataset. The pile-up correction is itself dependent on the incident photon spectrum, which of course the ARF, RMF and observed counts spectrum are used to determine. An iterative analysis procedure is suggested, using the **rmfgen** pile-up correction feature only. See the **rmfgen** description for details.

5.10 CCD gap and bad pixel corrections

By default the task will reduce the effective area produced in the ARF if the source area contains bad pixels/columns or contains area which lies outside the CCD boundaries. For an extended source this reduction is simply the ratio of the dead area within the source region to the good area. For point sources a fine grid is used to determine the contribution to the enclosed energy of each dead point, i.e. the dead regions are normalised by the fraction of the point spread function lying within that region before being subtracted from the total effective area. The task uses the parameter `extendedsource` to choose between these options. Although the correction for point sources is quite good the extended source correction assumes a uniform surface brightness of the source. If this condition is not met then quite large errors can be introduced. A way to avoid this potential source of error is to apply a detector mask in the accumulation of extended source spectra. The detector mask is a binary (1/0) detector image, where all CCD "dead" areas are set to 0. The masked out pixels are not used in the calculation of the effective area (see section 5.2.1). Hence, the integrated effective area over the extended source surface brightness takes into account only the fraction of the source which has been effectively imaged.
A detector mask is generated with **emask** starting from an exposure map as generated by **eexpmap**, which in turn requires an image generated by **evselect** or **xmmselect**. In command-line terms:

```
eexpmap imageset=field_of_view.img attitudeset=0060_0122700101_AttHk.ds
eventset=0060_0122700101_EPN_S001_ImagingEvts.ds
expimageset=exposure_map.fits pimin=1000 pimax=10000

emask expimageset=exposure_mask.fits detmaskset=detector_mask.fits
```

If the fraction of counts, lost to bad pixels and chip gaps is large, then it is recommended to use a finer resolution when determining this contribution. A warning is issued by the program if it sees that the resolution used is too coarse.

In a similar way the task corrects for area of the source region which is outside the FOV (selectable with the parameter `ignoreoutoffov`), outside the observing window or on a CCD explicitly excluded by a CCD selection within the data subspace.

This behaviour can be turned off by setting `withbadpixcorr=false`. CCD gaps and area outside the FOV are automatically corrected for if `withbadpixcorr=true`. To correct for bad pixels, a file containing bad pixel extensions must be specified on the command line by `badpixlocation`=*file*. The bad pixel information is usually stored in the event file from which the spectrum was constructed. Bad columns and bad rows (MOS only) are also handled if they are specified in the OFFSETS column of the

`badpixlocation` file. Pixels adjacent to chip gaps, bad pixels and bad columns are also corrected for if the FLAG selection in the datasubspace has been set accordingly. The code checks each bit mask in the event flag to see if pixels next to a particular type of bad pixel or bad column should be excluded. In this way spectra created with, for example, the #XMMEA_EM and #XMMEA_SM flagging can produce different effective areas for certain selection regions.

We recommend users to utilize #XMMEA_EM for accumulating EPIC-MOS spectra. Spectra extracted using this selection criteria have been demonstrated to be in agreement with the EPIC calibration status document (http://xmm2.esac.esa.int/docs/documents/CAL-TN-0018.pdf).

The task creates temporary files called BADPIXnn.ds, where nn is the CCD number, in the current directory while calculating the bad pixel correction. To make these files permanent specify `withfilteredset` = `yes` on the command line.

5.11 Out of time event correction

For EPIC-PN, in particular, there is a mode-dependent loss of photons due to the readout cycle. This can result in a 6% dead-time effect for full frame mode. This has been taken into account in the exposure time in the spectrum header for files created with pipeline versions greater than 20010618 but wasn't catered for in earlier pipeline runs. **arfgen** checks the PPSVERS keyword in the file header and applies the correction to the output ARF if the pipeline version is older than 18 June 2001 and if the parameter `modelootcorr` is set to true (the default). The PPSVERS keyword is written by pipeline modules and not by the SAS ∗proc routines; this means that if the data is reprocessed to create new event files then this keyword will not be present. If **arfgen** does not find the keyword it issues a warning. This should be ignored if the event file has been created with SAS 5.3 or any SAS version not older than 18 June 2001.

A PN ARF would be decreased by the following amount:

Mode	Factor (%)
PRIME_FULL_WINDOW	6.2
PRIME_FULL_WINDOW_EXTENDED	2.3
PRIME_SMALL_WINDOW	1.1

5.12 A cross-region ARF

Due to the finite size of the PSF in XMM, a fraction of photons which are emitted from one part of the sky are detected at another position on the detector. This is generally not a problem for point sources but for regions of extended emission it can cause an extracted spectrum to be a mix of flux from different areas of an object. It may be possible to disentangle the true sky emission by simultaneously fitting different spectral files. In this case an ARF giving the expected contribution (effective area) of flux received in one area from a different area is needed. This can be achieved by finding the product of the usual effective area components at the position of the output area with the energy-dependent encircled-energy fraction contained in the output area due to flux from the input area.

i.e.

$$A_{io} = F_o C_o Q_o \sum_i I_i \psi_{io} V_i / \sum_i I_i$$

where F_o is the filter response, C_o is the chip gap and bad pixel factor, Q_o is the quantum efficiency, I_i is the relative flux in a pixel i of the input region, ψ_{io} is the fraction of the PSF which falls in the output region from this input pixel and V_i is the vignetting at the position of the input region pixel.

6 Parameters

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

Parameter	Mand	Type	Default	Constraints
spectrumset	yes	dataset	spectrum.ds	none

Name of input spectrum dataset.

arfset	no	dataset	ancillaryfile.ds	none

Name of output ancillary response dataset.

withrmfset	no	boolean	false	none

If set, the ARF energy grid is defined by the RMF dataset specified by `rmfset`.

rmfset	no	dataset	responsefile.ds	none

Name of input response dataset. Used only if `withrmfset` is set to `true`.

detmaptype	no	choice	flat	dataset flat psf

The origin of the detector map: if set to `dataset`, then the user must specify the name of that dataset, and the table name of the map though `detmaparray`. If it is set to `flat` or `psf`, then the parameters `detxoffset`, `detyoffset`, `detxbins` and `detybins` define the detector map grid. In addition, `psfenergy` needs also to be set if `detmaptype` = `psf`.

detmaparray	no	array	detmapfile.ds:	none

Name of detector map dataset and array in the DAL compound notation. Only used of `detmaptype` is set to `dataset`.

withdetbounds	no	boolean	false	none

For `psf` or `flat` detector maps. If `true` then the bounds of the internal detector map are taken from the parameters `detxoffset` and `detyoffset`. Otherwise, the bounds are taken from the DSS information in the spectrum dataset.

detxoffset	no	real	1200	0 <= detxoffset <= 30000

Used if `detmaptype` is set to `psf` or `flat`. This defines the half-width of the detector map in DET coordinates. For example, if the source x-coordinate is x, the x-bounds are [x−`detxoffset`, x+`detxoffset`].

detyoffset	no	real	1200	0 <= detyoffset <= 30000

Used if `detmaptype` is set to `psf` or `flat`. This defines the half-height of the detector map in DET coordinates. For example, if the source y-coordinate is y, the y-bounds are [y−`detyoffset`,y+`detyoffset`].

withdetbins	no	boolean	false	none

For `psf` or `flat` detector maps. If `true` then the number of bins of the internal detector map are taken from the parameters `detxbins` and `detybins`. Otherwise, the number of bins are calculated automatically.

detxbins	no	integer	5	≥ 1

Used if `detmaptype` is set to `psf` or `flat`. This is the number of bins along the x-axis of the map.

detybins	no	integer	5	≥ 1

Used if `detmaptype` is set to `psf` or `flat`. This is the number of bins along the y-axis of the map.

psfenergy	no	real	2.0	$0 \leq$ psfenergy ≤ 30

Used if `detmaptype` is set to `psf`. This is the energy, in keV, at which the PSF model used to construct the detector map is taken.

filterdss	no	boolean	true	none

If `true`, the raw detector map and the bad pixel lists are filtered using the DSS information stored in the spectrum dataset. If `false`, every pixel in the raw detector map is propagated to the filtered map and every bad pixel is used in the area calculation. This is a debugging parameter and should not be set to false under normal conditions.

withfilteredset	no	boolean	false	none

If `true`, the filtered detector map is written out to a permanent dataset, the name of which is taken from the parameter `filteredset`.

filteredset	no	dataset	filteredpixellist.ds	none

Name of output filtered detector map. Used only if `withfilteredset` is set to `true`.

withsourcepos	no	boolean	false	none

If `true`, the source position must be specified via the parameters `sourcecoords`, `sourcex`, `sourcey`. Otherwise, the source position is taken from the centre of the source region defined in the data subspace of the input spectrum.

sourcecoords	no	choice	eqpos	eqpos pos tel det

Used if `withsourcepos = true`. The coordinate system for which the source position, specified by the parameter `sourcex` and `sourcey`, is defined. If `sourcecoords` is set to `eqpos`, then `sourcex, sourcey` correspond to RA and DEC respectively in decimal degrees. If `sourcecoords = pos`, then `sourcex, sourcey` correspond to POS coordinates. (Note that the POS coordinates are defined relative to a norminal pointing position; this is taken from the global attributes `REFXCRVL` and `REFYCRVL` of the spectrum dataset.) If `sourcecoords = tel`, then `sourcex, sourcey` correspond to the telescope coordinates theta (arcseconds) and phi (radians). If `sourcecoords = det`, then `sourcex, sourcey` are the x and y positions of the source centre, in DET coordinates.

sourcex	no	real	0	none

(Used if `withsourcepos = true`) The x-position of source centre, in terms of the coordinate system specified in `sourcecoords`.

sourcey	no	real	0	none

(Used if `withsourcepos = true`) The y-position of source centre, in terms of the coordinate system specified in `sourcecoords`.

extendedsource	no	boolean	false	none

Informs **arfgen** as to whether the source is extended or point-like. This is currently only used for the encircled energy correction.

useodfatt	no	boolean	false	none

whether the full attitude file should be used from the OAL. If this is set to false then attitude is taken from the mean pointing position in the spectrum header.

withbadpixcorr	no	boolean	true	none

states whether CCD gap and bad pixel corrections are wanted.

badpixlocation	no	dataset	"notSpecified"	none

The name of the file containing the bad pixel locations. Usually, this will be the event file from which the spectral dataset was constructed.

badpixelresolution	no	real	0.5	none

This is the resolution that the program uses to calculate the encircled energy correction and to search for pixels which are outside the field-of-view or off the edge of the CCDs. If the `extendedsource` parameter is set *true* then this value is defaulted to 2.0 arcseconds. Reducing the number, in this case, to 1.0 arcseconds will improve the accuracy of the calculation at the expense of execution time.

ignoreoutoffov	no	boolean	yes	

Whether area outside the field of view should be excluded in the arfgen calculation.

modeleffarea	no	boolean	true	none

If `true`, the effective area data from the CAL is taken into consideration when generating the ARF.

modelquantumeff	no	boolean	true	none

If `true`, the quantum efficiency data from the CAL is taken into consideration when generating the ARF.

modelfiltertrans	no	boolean	true	none

If `true`, the filter transmission data from the CAL is taken into consideration when generating the ARF.

modelee	no	boolean	true	none

If it is point-like (`extendedsource`=false), a model of the PSF positioned at a point specified by `sourcex` and `sourcey` is generated, and the fraction of the model situated within the region defined by the DSS information is computed.

There is currently no encircled energy correction implementation for extended sources.

psfmodel	no	string	ELLBETA	ELLBETA,EXTENDED,HIGH

The point spread function (PSF) model to use when calculating the encircled energy correction.

modelootcorr	no	boolean	true	none

The ARF will be decreased by a mode-dependent percentage if `modelootcorr`=true and if the pipeline version used to create the spectrum is older than 18 June 2001.

eegridfactor	no	integer	50	≥ 1

Used if `modelee` is set to `true`. This is the factor which to oversample the ARF energy grid in order to perform the energy correction efficiently: The larger the factor, the faster the execution time, but the less accurate the correction.

setbackscale	no	boolean	false	none

states whether the BACKSCAL value in the input spectrum should be set to the calculated source region area.

keeparfset	no	boolean	true	none

whether the calculated ARF file should be kept or deleted. This option is included for the benefit of the **backscale** metatask which sets the BACKSCAL value in the input spectrum.

crossregionarf	yes	boolean	false	none

whether across-region ARF should be calculated.

crossreg_spectrumset	yes	dataset		none

Name of spectrum containing the input region.

7 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.

invalidInstrument *(error)*
The value of the `INSTRUME` attribute of the spectral file is not EMOS1, EMOS2 or EPN. (This indicates a bug - raise an SPR)

invalidArraySize *(error)*
The detector map array is not 2-dimensional as expected.

invalidAreaEnergyGrids *(error)*
The number of elements in the computed area vector do not match that in the energy grid. This indicates a bug - raise an SPR.

mapXRangeInvalid *(error)*
The requested X-range for the flat or psf detector map arrays is not such that `detxmin` >= `detxmax`. (This indicates a bug, so an SPR should be raised).

mapYRangeInvalid *(error)*
The requested Y-range for the flat or psf detector map arrays is not such that `detymin` >= `detymax`. (This indicates a bug, so an SPR should be raised).

mapSizeInvalid *(error)*
The number of pixels specified for the flat or psf detector maps is zero (This indicates a bug, so an SPR should be raised).

InvalidSourcePosition *(error)*
The source does not lie within the region corresponding to that of the detector map.

sourceCoordTypeInvalid *(error)*
The value of the parameter `sourcecoords` is not a valid type (this indicates a bug - raise an SPR)

NoPatterns *(error)*
There are no patterns in the internal list (this indicates a bug - raise an SPR.)

InvalidWCSAxes *(error)*
The detector map array WCS axis types were not DETX and DETY.

DSSNotFound *(error)*
The input spectrum does not contain a *mandatory* data subspace component. Arfgen needs this information to determine the position of the source on the detector among other things. Ensure that the writedss option is set true within the evselect parameters

RegionInvalid *(error)*
The data subspace in the input spectrum either doesn't contain region information or it is unbounded in detector coordinates.

ShapeNotHandled *(error)*
The data subspace contains a shape which is not recognised by arfgen

NoRmfTablePresent *(error)*
The rmf file specified doesn't contain an RMF table from which the energies can be extracted

ArfObjError *(error)*
An error has occurred within arfgen which is not understood. The most likely reason is that the software environment or the input datafile is corrupt in some way. Try redefining the SAS environment, including CCF directories and regenerate the spectrum.

BadFilterName *(error)*
The Filter keyword in the spectrum header is either missing or doesn't match a known filter

negativeDetmapPixel *(warning)*
A pixel within the detector map has a negative value.
corrective action: Continue processing - the pixel would provide negative weighting to the total ARF

regUnbounded *(warning)*
The selected regions are unbounded along the positive or negative directions along either the DETX, DETY, X or Y axes.
corrective action: continue

detmapXboundsExceeded *(warning)*
The extent of the selected region projected along the DETX axis exceeds the detector map bounds for that axis.
corrective action: continue

detmapYboundsExceeded *(warning)*
The extent of the selected region projected along the DETY axis exceeds the detector map bounds for that axis.
corrective action: continue

InvalidPosWCSInfo *(warning)*
One or more of the global attributes `REFXCTYP`, `REFXCRPX`, `REFXCRVL`, `REFXCDLT`, `REFYCTYP`, `REFYCRPX`, `REFYCRVL`, `REFYCDLT` are not present in the spectrum dataset.
corrective action: Provide a dummy WCS setting, for the POS coordinate system and continue. The RA/DEC reference pointing, corresponding to `REFXCRVL` and `REFYCRVL`, is instead taken from the attributes `RA_PNT` and `DEC_PNT`

zeroSumDetmap *(warning)*
This means that the detector map has all zero elements. The resultant ARF will be full of zeroes. This is unimportant if **arfgen** has been called from **backscale** but is a major error if **arfgen** is being used standalone. If a 'flat' or 'psf' map has been used it may be that

more elements are needed to fully describe the source region. An alternative possibility is that the attitude reconstruction is bad for this observation.
corrective action: continue

zeroRenorm *(warning)*
Similar to 'zeroSumDetmap' but may mean that no detector map pixels lie within the boundaries of a CCD. This can occur if the backscale calculation is performed on a source region whose centre lies on a chip gap.
corrective action: continue

NoBadPixLocations *(warning)*
The bad pixel file hasn't been specified on the command line or is invalid. This means that no correction for bad pixels will be applied
corrective action: continue

EEoutofBounds *(warning)*
The circular PSF model is valid for energies upto 9keV for the MOS and 10.5 keV for EPIC-PN. This warning reminds the user that the higher energies in the ARF will be corrected for the encircled energy fraction using the 9keV PSF, which is at best an approximation
corrective action: continue

EnergyOutsideValidityRange *(warning)*
This message comes from the CAL package and normally means that the PSF is being evaluated outside its standard range. This is usually harmless
corrective action: continue

OffaxisAngleOutsideValidityRange *(warning)*
This means that a quantity, usually the PSF, is being evaluated at an off-axis angle outside those at which the quantity has been calibrated. The quantity relevant to the furthest off-axis calibration point is returned.
corrective action: continue

NoPPSVersion *(warning)*
The input spectrum doesn't contain the pipeline processing version. This means that the s/w doesn't know whether it should apply the out-of-time correction or not. See section 5.11
corrective action: continue without applying out-of-time correction.

NonStandardPatterns *(warning)*
The pattern fractions used to scale the detector Quantum Efficiency are defined for a set number of standard pattern ranges (0, 0-4, 1-4, 0-12). The task can not cope with a different pattern range and so defaults to using the full set of patterns
corrective action: assume that the spectrum contains the full range of patterns

UnknownModeString *(warning)*
If the spectrum contains an observing mode (in the keyword SUBMODE) which is not recognised then the software assumes that the common PrimeFullWindow mode was in use. The mode is used to calculate certain calibration quantities such as the pattern fractions
corrective action: PrimeFullWindow

BadModePatternCombination *(warning)*
This means that the pattern selection in the input spectrum is not calibrated for the observing mode. e.g. Any pattern selection other than 0-4 will cause this warning to be issued for PN timing mode spectra.
corrective action: recreate the spectrum using the recomended pattern range

NegBackscale *(warning)*
The area of the spatial region (BACKSCAL) has been calculated to be negative. This can happen and usually means that the chosen area has no good pixels due to it being outside

the field-of-view or obliterated by bad pixels, bad columns or chip gaps. The program sets the BACKSCAL value to zero in the output file.
corrective action: 0

8 Input Files

1. Input OGIP-compliant spectral dataset. The structure of an OGIP-compliant spectrum is detailed in [3], but for the purposes of **arfgen** the dataset must contain the following information:

   - The global attributes
     - DATE-OBS
     - INSTRUME
     - TELESCOP
     - DATAMODE
     - FILTER
     - RA_PNT
     - DEC_PNT
     - PA_PNT
     - REFXCTYP
     - REFXCRPX
     - REFXCRVL
     - REFXCDLT
     - REFYCTYP
     - REFYCRPX
     - REFYCRVL
     - REFYCDLT
   - A table named SPECTRUM. If the user has generated a spectrum over a selected region or regions, this table must also contain DSS information corresponding to those regions. This information can be added by **evselect** to the spectrum during its creation by setting the **evselect** parameter writedss to true.
   - The above also applies for the cross-region spectrumset.

2. Input detector map. This is an array specified by the parameter detmaparray and containing the following information:

   - A 2-d array of any numerical type. All pixels within the array must exceed zero, and the sum of all pixels must also exceed zero.
   - The WCS attributes CTYPE1, CRPIX1, CRVAL1, CDELT1, CTYPE2, CRPIX2, CRVAL2, CDELT2. This map must be generated in terms of [DETX,DETY] coordinates, so the axis types, ie CTYPE1 and CTYPE2, must be DETX and DETY respectively.

   This map is only used by **arfgen** if detmaptype is set to 'dataset'.

3. A standard EPIC event file (optional)

9 Output Files

1. An OGIP-compliant Ancillary Response File, as defined in [4].

   This contains a table labelled `ARF`, that in turn contains:

   - A column labelled `ENERG_LO`, containing the lower bounds for each energy bin (keV).
   - A column labelled `ENERG_HI`, containing the upper bounds for each energy bin (keV).
   - A column labelled `SPECRESP`, listing the area for each energy bin described by the `ENERG_LO` and `ENERG_HI` columns.

   and the following attributes:

   - `TELESCOP`
   - `INSTRUME`
   - `FILTER`
   - `ARFVERSN`
   - `HDUCLASS`
   - `HDUCLAS1`
   - `HDUCLAS2`
   - `HDUCLAS3`
   - `HDUVERS1`

2. A intermediate and temporary dataset containing the input detector map in pixel list form. used as input to **attcalc** to compute the detector-sky coordinate conversion.

   This contains a table labelled `EVENTS`, which in turn contains:

   - A column labelled `VALUE` of type real-64
   - A column labelled `X` of type int-32
   - A column labelled `Y` of type int-32
   - A column labelled `DETX` of type int-16
   - A column labelled `DETY` of type int-16
   - A column labelled `TIME` of type real-64

   and the following attributes:

   - `TELESCOP`
   - `INSTRUME`
   - `DATAMODE`

3. A version of the input detector map, in pixel list form, filtered using the DSS information stored in the spectrum dataset. This is written out as a permanent dataset with a name corresponding to `filteredset` if `withfilteredset` is set to `true`.

   This contains a table labelled `EVENTS`, which in turn contains:

   - A column labelled `VALUE` of type real-64
   - A column labelled `X` of type int-32
   - A column labelled `Y` of type int-32
   - A column labelled `DETX` of type int-16
   - A column labelled `DETY` of type int-16

- A column labelled TIME of type real-64

and the following attributes:

- TELESCOP
- INSTRUME
- DATAMODE

10 Algorithm

10.1 Initialisation

- From parameter settings and the relevant attributes in the spectral file, set the state of the arfgen data server before processing is performed. For example, if a detector map is specified, read in that detector map. If a particular factor is requested to be included in the ARF by the user, set the data server (ie create the relevant objects) to provide that correction. Also perform various checks, such as ensuring that the correct global attributes are present in the spectrum dataset, and ensuring that the extents of the selected regions lie inside the bounds of the detector map.
- Convert the input detector map into a pixel list, and filter this using the Data SubSpace information stored in the input spectum dataset as detailed below. The result is an internal, filtered detector map that is used in the subsequent processing.
  - From the detector map, generate a table containing the columns DETX, DETY and VALUE, describing the DET coordinates and value for each pixel in the detector map.
  - Add a dummy TIME column (required by attcalc, although not used directly, as attcalc is called with attitudelabel = ‘user‘.)
  - Run attcalc on table, generating sky (X, Y) columns
  - Get the DSS information from the spectrum dataset, and ’project’ this information onto the planes that contain the POS and the DET axes pairs.
  - filter the table using the projected DSS information.

10.2 Main stage: Generate area

- Compute the source position in DET coordinates [xdet, ydet] from sourcex, sourcey: If the input coordinate system (form sourcecoords) is POS, convert sourcex, sourcey to EQPOS, using the attributes REFXCRVL and REFYCRVL from the spectrum as the reference point, then convert to DET using the spacecraft attitude at the start of the observation, taken from the OAL. If the coordinate system is EQPOS, convert to DET coordinates as above. If the parameter withsourcepos is false, then get the RA and DEC of the target from the attributes RA_PNT, DEC_PNT and convert to DET coordinates as with EQPOS above.
- Use xdet, ydet to determine the source position in TELCOORDs and compute mirror_area for that position.
- Compute sum of pixel values in the filtered detector map (detmap_sum).
- foreach detector pixel within the filtered detector map, across all CCDs

  - Compute quantum efficiency curve (as a function of energy) for all selected patterns
  - foreach mean energy value
    * Compute the area for that pixel at that energy, in accordance to the following formula:

$$\begin{aligned} \texttt{eff\_area\_pixel} \;=\; & \texttt{mirror\_area} \times \\ & \texttt{filter\_transmission[pixel, energy]} \times \\ & \texttt{quantum\_efficiency[pixel, energy]} \times \end{aligned} \tag{1}$$

      The filter_transmission and quantum_efficiency values for each pixel are provided by the calibration data selected. The mirror area includes vignetting: for point sources this is calculated once at the source position, whereas for extended sources it is computed at the position of each detector map pixel.

    * multiply product by the detector map pixel value.
    * add result for pixel calculated above to running total at that energy
  - end energy loop
- end pixel loop
- divide resultant area vector by detmap_sum
- correct area vector for encircled energy
- correct area vector for bad pixels/columns and CCD gaps

10.3 Encircled Energy Correction

- Rebin the energy grid used in the main stage by a factor eegridfactor
- For each energy
  - Generate a model of the PSF centred at [xdet,ydet].
  - Loop over each cell in a grid, with resolution given by badpixelresolution, calculating the PSF contribution in every cell. Add the cell contribution to a running total which finally gives the fraction of the PSF within the selected region for this energy
  - Write encircled energy fraction as an element of an encircled energy fraction vector.
- Interpolate EE fraction vector to map to original energy grid used in main stage.
- Divide area vector by EE fraction vector.

10.3.1 Cross region ARF

- Create a filtered detector map for both the input and output regions from the input image.
- for each detector map pixel in the output region
  - foreach image pixel in the input area
    * Calculate fraction of PSF from this input pixel which lands in this output pixel

11 Comments

- Dead time corrections are not considered explicitly in this task, as the `EXPOSURE` attribute, written to the spectral file by **evselect** takes into account dead time effects.
- **arfgen** calls the task **attcalc** internally. The command line equivalent to this call is
  ```
  attcalc eventset=filteredtable.ds attitudelabel=fixed
  fixedra=RA_PNT fixeddec=DEC_PNT fixedposangle=PA_PNT
  refpointlabel=user nominalra=REFXCRVL nominaldec=REFYCRVL withatthkgen=false
  ```

  where `filteredtable.ds` corresponds to the intermediate event dataset generated by **arfgen** for the purpose of generating a filtered detector map. The labels in capitals mark global attributes from the spectrum dataset.

  Note that the intermediate event dataset also contains attributes used to initialise the CAL in the subsequent call by **attcalc**. In order for the CAL to be initialised completely, the following global attibutes are set temporarily as follows:

  For MOS1/2:

  - `CCDID = 1`
  - `CCDNODE = 0`
  - `DATATYPE = "IMAGE.EL"`
  - `WINDOWX0 = 0`
  - `WINDOWY0 = 0`
  - `WINDOWDX = 610`
  - `WINDOWDY = 602`
  - `EDUMODE = 3`
  - `FRMTIME = 26`

  For PN:

  - `QUADRANT = 0`
  - `CCDID = 0`
  - `CCDNODE = 0`
  - `CCDMODE = 1`
  - `DATATYPE = "IMAGE.EL"`
  - `WINDOWX0 = 0`
  - `WINDOWY0 = 1`
  - `WINDOWDX = 64`
  - `WINDOWDY = 199`
  - `EDUMODE = 3`
  - `FRMTIME = 73`

  These settings are based on the values found in the ODF files `0070_0123700101/0070_0123700101_M1S00110IME.` (MOS1) and `0070_0123700101/0070_0123700101_PNS00301IME.FIT` (PN) at the time of writing (14 Oct 2000).

  These will be replaced in future versions of **arfgen** with the correct values for each CCD for the detector in question.
- The pile-up correction feature is only an initial version which is not maintained. It is strongly recommended that the user should not use this feature.

References

[1] Christian Erd, Phillipe Gondoin, David Lumb, Rudi Much, Uwe Lammers, and Giuseppe Vacanti. Calibration Access and Data Handbook. Technical Report XMM-PS-GM-20, ESA/SSD, Jan 14 2000. Found at the URL: `http://xmm.vilspa.esa.es/docs/documents/CAL-MAN-0001-2-1.ps.gz`.

[2] Ballet J. Pile-up on x-ray ccd instruments. *Astron. Astrophys. Suppl.*, 135:371–381, 1999.

[3] I.M. George K.A. Arnaud and A.F. Tennant. The OGIP Spectral File Format. Technical Report OGIP/92-007, NASA/GSFC, Sept 1992. Found at the URL: `http://legacy.gsfc.nasa.gov/docs/heasarc/ofwg/docs/summary/ogip_92_007_summary.html`.

[4] I.M. George K.A. Arnaud and A.F. Tennant. The Calibration Requirements for Spectral Analysis. Technical Report OGIP/92-002, NASA/GSFC, Dec 1998.

[5] 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.