backscale

August 16, 2005

Abstract

A tool for calculating and writing the BACKSCAL keyword in EPIC spectra.

1 Instruments/Modes

Instrument	Mode
EPIC	SPECTROSCOPY

2 Use

pipeline processing	yes
interactive analysis	yes

3 Description

This task calculates the area of a source region used to make a spectral file. The area is written into the header of the SPECTRUM table of the file in the keyword BACKSCAL.

The final value is:

```
area = geometric_area - ccd_gaps - bad_pixels
```

In normal use only pixels which lie within the CCD boundaries and not on bad pixels contribute to the total area. The units of area are *detector pixels* which are square pixels of side 0.05 arcseconds.

If the parameter `withbadpixcorr` is set false then the pure geometric area will be calculated regardless of where the source region lies. If `withbadpixcorr` is true then pixels lying off the edges of all the CCDs will be subtracted from the total area. If `withbadpixcorr` is true and `badpixlocation` is set to a file containing bad pixel extensions (typically the input event file) then bad pixels lying within the source region area are also subtracted from the total area. By default, area outside the field of view is not included in the backscale alculation. This can be overridden by setting `ignoreoutoffov` false on the command line.

4 Parameters

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

Parameter	Mand	Type	Default	Constraints
spectrumset	yes	string	spectrum.ds	

Name of the input file

badpixlocation	no	string	notSpecified	

Name of the file containing the bad pixels, initially this is the event file.

withbadpixcorr	no	boolean	yes	

Whether to use bad pixels and chip gaps in the calculation.

useodfatt	no	boolean	no	

Whether to use the ODF attitude file to construct position info.

ignoreoutoffov	no	boolean	yes	

Whether area outside the field of view should be included in the backscale calculation.

5 Errors

There are no errors specific to **backscale**. See the **arfgen** documentation for a list of its internal errors.

NB: A lot of warnings may be generated by arfgen when being run in this backscale calulation mode. These may look worrying but are ALL irrelevant and should be ignored !

6 Input Files

- an EPIC spectrum file containing a datasubspace definition
- an optional second file containing the bad pixel extensions

7 Output Files

- The input spectrum is modified

8 Algorithm

```
Create a grid which encompasses the source region
```

```
if (source region half-width < 2 arcmins) {
   grid_element_Width = 10
}
else {
   grid_element_Width = 20
}

area=0
Loop over each grid element
{
   if (element lies within a CCD) {
      area = area + grid_element_Width * grid_element_Width
   }
}
Loop over each bad pixel
{
   if (bad pixel lies within source region) {
      area = area - bad_pixel_area_in_detector_pixels
   }
}
```

9 Comments

To improve the execution time of the task the source region is divided into a grid which is used to check whether pixels lie on a CCD. The width of each grid element is set to be 10 detector pixels ($\sim$ 0.5 arcseconds) for small regions (region half-width less than 2 arcminutes) and 20 detector pixels ($\sim$ 1 arcsecond) for larger regions. It has been chosen like this to give good coverage of CCD gaps where they make a significant contribution to the total area while keeping reasonable execution times for larger areas. This results in an execution time of $\sim$1 minute for a circular source region of 1 arcminute radius and $\sim$5 minutes for a circle of radius 5 arcminutes.

References