rgsenergy

April 16, 2023

Abstract

This task performs energy calibrations on RGS telemetry. The input dataset is an event list created by **rgsframes** and is modified by the addition of the `PI` column to the `PIXELS` table.

1 Instruments/Modes

Instrument	Mode
RGS	Spectroscopy
RGS	High Time Resolution

2 Use

pipeline processing	yes
interactive analysis	yes

3 Description

This task calibrates the telemetered RGS pixel energies. It operates on a CCD-specific raw event list created by task **rgsframes** and adds the `PI` column to the `PIXELS` table. The calibration consists of three corrective procedures performed in sequence: offset correction, gain correction, and CTI correction. The CAL Handbook describes each procedure in detail. The offset correction can use static calibration data from the CCF (one value per CCD and node by each RGS), but also dynamic offset data derived from the diagnostic images can be used for subtraction. There are two methods available:

- Using averaged diagnostic images derived from three consecutive orbits, including the one corresponding to the data (parameter `withdiagoffset`). In this case the subtraction is performed pixel by pixel. This is the method considered as most accurate, due to the offset modulations observed with time within a CCD and node.
- Using any diagnostic data for derivation of one value per CCD and node (parameter `withoffset`).

The method used for offset correction is flagged in the keyword "OFFSCORR" in each EXPOSURE extension of the events file (either as "CCDNODE" or "PIX2PIX", corresponding to the one value or pixel by pixel subtraction, respectively).

For diagnostic purposes the gain and CTI corrections can be disabled independently (parameters `withgain` and `withcti`).

Throughout this document a † marks items that do not apply to HTR mode data and a ‡ marks items that apply only to HTR mode data.

4 Parameters

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

Parameter	Mand	Type	Default	Constraints
ccdset	yes	dataset	events.ds	

A CCD-specific event list from **rgsframes**.

withoffset	no	boolean	no	

Enables override of the static energy offset calibration data, using a data file (parameter `offset`) such as produced by task **rgsoffsetcalc**.

offset	no	dataset	offset.ds	

Energy offset calibration override data, enabled by parameter `withoffset`.

withdiagoffset	no	boolean	yes	

Performs the offset calibration by subtracting pixel by pixel offset values obtained from averaging diagnostic data over three consecutive orbits, including the one corresponding to the data. The average diagnostic offset files (rrrr_ooooooooo_R[1][2]X000000OFX.FIT) have to be included in the ODF.

withgain	no	boolean	yes	

Enables gain correction.

withcti	no	boolean	yes	

Enables CTI correction.

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.

6 Input Files

- `ccdset`

  The following tables are required from the input dataset, which is expected to have been created by **rgsframes**.

  - `EXPOSURE`

    Required columns:

`FRAME`	int32	frame identifier
`TIME`	real64	timestamp

  - `PIXELS`

    Required attributes:

`WINDOWX0` `WINDOWY0`	telemetry window: origin (node coordinates)
`WINDOWDX` `WINDOWDY`	telemetry window: dimensions
`RAWY`‡	nominal readout row in HTR mode

    Required columns:

`FRAME`	int32	frame identifier
`RAWX` `RAWY`†	int16	coordinates
`ENERGY`	int16	uncalibrated energy
`CCDNODE`	int8	node identifier
`SHAPE`†	int8	shape code
`GRADE`†	int8	number of pixels

- `offset`

  CCF component containing RGS energy offset correction override data. See **rgsoffsetcalc** for further details.

7 Output Files

- `ccdset`

  The input dataset is modified by the addition of one column to the `PIXELS` table:

`PI`	real32	calibrated energy

8 Algorithm

```
temporary column: TIME = EXPOSURE:FRAME[0]
FOR EACH i,j such that (FRAME[i] == EXPOSURE:FRAME[j]):
  TIME[i] = EXPOSURE:TIME[j]

new column: PI
IF (datamode == Spectroscopy):
  PI = EnergyCorrector::offsetCorrect(TIME,RAWX,RAWY,SHAPE,CCDNODE,ENERGY)
ELSE:
  PI = EnergyCorrector::offsetCorrect(TIME,RAWX,CCDNODE,ENERGY)

IF (*withgain):
  PI = EnergyCorrector::gainCorrect(TIME,CCDNODE,PI)

IF (*withcti):
  temporary column: SEP = 1000
  IF (datamode == Spectroscopy):
    PI = CtiCorrector::ctiCorrect(TIME,RAWX,RAWY,SHAPE,SEP,CCDNODE,PI)
  ELSE:
    PI = CtiCorrector::ctiCorrect(TIME,RAWX,RAWY,CCDNODE,PI)
```

9 Comments

References